

TUGAS AKHIR - KI141502

Segmentasi Trabecular Bone Dental Panoramic Radiograph Berbasis Karakteristik Profil Segmen Menggunakan Extreme Learning Machine

RIZQI OKTA EKOPUTRIS
5113100005

Dosen Pembimbing I
Dr. Agus Zainal Arifin, S.Kom., M.Kom.

Dosen Pembimbing II
Arya Yudhi Wijaya, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

**SEGMENTASI TRABECULAR BONE DENTAL
PANORAMIC RADIOGRAPH BERBASIS
KARAKTERISTIK PROFIL SEGMENTASI
MENGUNAKAN EXTREME LEARNING MACHINE**

**RIZQI OKTA EKOPUTRIS
5113100005**

**Dosen Pembimbing I
Dr. Agus Zainal Arifin, S.Kom., M.Kom.**

**Dosen Pembimbing II
Arya Yudhi Wijaya, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

**TRABECULAR BONE SEGMENTATION BASED ON
SEGMENT PROFILE CHARACTERISTICS USING
EXTREME LEARNING MACHINE ON DENTAL
PANORAMIC RADIOGRAPHS**

RIZQI OKTA EKOPUTRIS
5113100005

Supervisor I
Dr. Agus Zainal Arifin, S.Kom., M.Kom.

Supervisor II
Arya Yudhi Wijaya, S.Kom., M.Kom.

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
Sepuluh Nopember Institute of Technology
Surabaya 2017

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

SEGMENTASI TRABECULAR BONE DENTAL PANORAMIC RADIOGRAPH BERBASIS KARAKTERISTIK PROFIL SEGMENT MENGGUNAKAN EXTREME LEARNING MACHINE

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Cerdas Visual
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

RIZQI OKTA EKOPUTRIS

NRP: 5113 100 005

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Dr. Agus Zainal Arifin, S.Kom., M.Kom.
NIP. 19720809 199512 1 001

Arya Yudhi Wijaya, S.Kom., M.Kom.
NIP. 19840904 201012 1 002



SURABAYA
JANUARI 2017

[Halaman ini sengaja dikosongkan]

**SEGMENTASI TRABECULAR BONE DENTAL
PANORAMIC RADIOGRAPH BERBASIS
KARAKTERISTIK PROFIL SEGMENT MENGGUNAKAN
EXTREME LEARNING MACHINE**

Nama Mahasiswa : Rizqi Okta Ekoputris
NRP : 5113 100 005
Jurusan : Teknik Informatika FTIF ITS
Dosen Pembimbing 1 : Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Dosen Pembimbing 2 : Arya Yudhi Wijaya, S.Kom., M.Kom.

ABSTRAK

Dental panoramic radiograph adalah citra x-ray dua dimensi (2-D) dari gigi yang merekam keseluruhan mulut, termasuk gigi, rahang atas, rahang bawah dan jaringan serta struktur yang melingkupinya dalam satu citra. Pada dental panoramic radiograph mengandung banyak informasi yang dapat diidentifikasi salah satunya melalui struktur trabecular bone.

Penelitian ini mengusulkan segmentasi area trabecular bone pada dental panoramic radiograph berbasis karakteristik profil segmen dan metode klasifikasi Extreme Learning Machine. Input dari metode ini adalah dental panoramic radiograph. Pemilihan region of interest (ROI) dilakukan pada tulang rahang bawah pada area trabecular bone yang didalamnya terdapat gigi dan cortical bone. ROI tersebut dibagi lagi menjadi dua dimana ROI atas mengandung gigi dan ROI bawah mengandung cortical bone. Setelah itu, hasil pemotongan ROI dilakukan preprocessing menggunakan filter mean dan median untuk ROI atas dan filter motion blur untuk ROI bawah. Citra yang telah terpisah tersebut masing-masing diekstrak tiap pikselnya menjadi 4 fitur yang terdiri dari intensitas citra, filter Gaussian 2D dengan dua sigma yang berbeda, dan filter Log Gabor untuk ROI atas. Untuk ROI bawah, digunakan 5 ekstraksi fitur yaitu intensitas citra, filter Gaussian 2D dengan dua sigma yang berbeda, phase congruency, dan Laplacian of Gaussian. Kemudian digunakan beberapa

sampel piksel sebagai data training untuk membuat model Extreme Learning Machine. Output dari classifier ini adalah area segmentasi dari trabecular bone.

Pada ROI atas, didapatkan rata-rata sensitivitas, spesifisitas, dan akurasi masing-masing sebesar 82,31%, 93,67%, dan 90,33%. Sedangkan pada ROI bawah didapatkan rata-rata sensitivitas, spesifisitas, dan akurasi masing-masing sebesar 95,01%, 96,50%, 95,29% dan 2,59 detik.

Kata kunci: dental panoramic radiograph, Extreme Learning Machine, Trabecular bone, segmentasi

TRABECULAR BONE SEGMENTATION BASED ON SEGMENT PROFILE CHARACTERISTICS USING EXTREME LEARNING MACHINE ON DENTAL PANORAMIC RADIOGRAPHS

Student's Name : Rizqi Okta Ekoputris
Student's ID : 5113 100 005
Department : Informatics Engineering, FTIF-ITS
First Advisor : Dr. Agus Zainal Arifin, S.Kom., M.Kom.
Second Advisor : Arya Yudhi Wijaya, S.Kom., M.Kom.

ABSTRACT

Dental panoramic radiograph is the x-ray image of two-dimensional (2-D) of the dental record an entire mouth, including the teeth, upper jaw, lower jaw and surrounding tissues and structures in the image. In dental panoramic radiograph contains a lot of information that can be identified one of them through the trabecular bone structure.

This research proposes the segmentation of trabecular bone area on dental panoramic radiographs based segment profile characteristics and classification methods Extreme Learning Machine. Input from these methods is dental panoramic radiographs. Selection of region of interest (ROI) was performed on the lower jawbone on trabecular bone area in which there are teeth and cortical bone. The ROI is divided into two where the upper ROI containing the teeth and lower ROI contains cortical bone. After that, the result of cutting ROI done preprocessing using mean and median filter to the upper ROI and motion blur for lower ROI. The image that has separated each extracted each pixel into 4 features consisting of image intensity, 2D Gaussian filter with two different sigma, and Log Gabor filter for upper-ROI. For lower ROI, used 5 feature extraction namely the intensity image, 2D Gaussian filter with two different sigma, phase congruency, and Laplacian of Gaussian. Then use some of the sample pixels as

training data to create models of Extreme Learning Machine. The output of this classifier is a segmentation of trabecular bone area.

On top ROI, obtained an average sensitivity, specificity, and accuracy of respectively 82.31%, 93.67%, and 90.33%. While at lower ROI obtained an average sensitivity, specificity, and accuracy of respectively 95.01%, 96.50%, 95.29% and 2.59 seconds.

Keywords: dental panoramic radiograph, Extreme Learning Machine, trabecular bone, segmentation

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “*Segmentasi Trabecular Bone Dental Panoramic Radiograph berbasis Karakteristik Profil Segmen menggunakan Extreme Learning Machine*”.

Buku Tugas Akhir ini disusun dengan harapan dapat memberikan manfaat dalam penelitian *dental panoramic radiograph* dan pengembangan ilmu pengetahuan. Selain itu, memberikan kontribusi positif bagi kampus Teknik Informatika, ITS.

Dalam perancangan, pengerjaan dan penyusunan Tugas Akhir ini, penulis banyak mendapatkan bantuan dari berbagai pihak. Tanpa mengurangi rasa hormat, penulis ingin mengucapkan terima kasih kepada.

1. Orang tua penulis Bapak Drs. Sutrisno Hadi dan Ibu Sri Puji Astutik yang telah memberikan dukungan moral, spiritual dan material serta selalu memberikan doa demi kelancaran penulis dalam mengerjakan tugas akhir.
2. Bapak Dr. Agus Zainal Arifin, S.Kom., M.Kom. selaku dosen pembimbing penulis yang telah memberikan ide, arahan dan motivasi sehingga penulis dapat menyelesaikan tugas akhir dengan tepat waktu.
3. Bapak Arya Yudhi Wijaya, S.Kom., M.Kom. selaku dosen pembimbing penulis yang telah memberikan nasihat dan perhatian sehingga penulis dapat menyelesaikan tugas akhir dengan tepat waktu.
4. Tohari Ahmad, S.Kom.,MIT.,Ph.D selaku dosen wali yang telah memberikan nasihat dan motivasi selama masa perkuliahan.

5. Ibu dan Bapak dosen yang sudah membagikan ilmunya kepada penulis selama masa perkuliahan Teknik Informatika ITS.
6. Ibu dan Bapak staf Tata Usaha yang telah memberikan bantuan dan kemudahan kepada penulis selama masa perkuliahan di Teknik Informatika ITS.
7. Titing Reza Fahriza yang menjadi motivator saya selama pengerjaan tugas akhir ini di Surabaya.
8. Kelompok Visi, Iham Gurat Adilion yang telah banyak membantu penyelesaian tugas akhir ini.
9. Teman-teman Admin KCV yang merupakan keluarga kedua penulis saat di Surabaya.
10. Rekan-rekan seperjuangan TC2013, yang selalu memberikan bantuan, semangat, canda tawa serta menemani penulis selama menempuh pendidikan di Teknik Informatika ITS.

Penulis menyadari masih ada kekurangan dalam penyusunan tugas akhir ini. Penulis mohon maaf atas kesalahan, kelalaian maupun kekurangan dalam penyusunan tugas akhir ini. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan kedepan.

Surabaya, Januari 2017

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
KODE SUMBER	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Tugas Akhir	3
1.5 Manfaat Tugas Akhir	3
1.6 Metodologi	3
1.7 Sistematika Laporan.....	5
BAB II DASAR TEORI.....	7
2.1 Dental Panoramic Radiograph dan Struktur Tulang	7
2.1.1 Pengertian Dental Radiograph	8
2.1.2 Teori Panoramic Radiographs.....	9
2.1.3 Struktur Susunan Dental Panoramic Radiograph.....	11
2.1.4 Struktur Tulang	11
2.1.5 Cortical bone	12
2.1.6 Trabecular bone	12
2.2 Filter	13
2.2.1 Normalisasi	13
2.2.2 Filter Median.....	14
2.2.3 Filter Mean.....	15
2.2.4 Filter Motion Blur.....	15
2.3 Ekstraksi fitur	15
2.3.1 Filter Gaussian 2D	16
2.3.2 Filter Log Gabor	17
2.3.3 Phase Congruency.....	21
2.3.4 <i>Laplacian of Gaussian (LoG)</i>	23

2.4	Metode Klasifikasi <i>Supervised: Extreme Learning Machine</i> (ELM).....	23
2.5	Penghitungan Performa	25
BAB III ANALISIS DAN PERANCANGAN		27
3.1	Tahap Analisis.....	27
3.1.1	Deskripsi Umum	27
3.1.2	Arsitektur Perangkat Lunak	27
3.1.3	Spesifikasi Kebutuhan Perangkat Lunak.....	28
3.1.4	Analisis Aktor	28
3.1.5	Analisis Permasalahan	28
3.2	Tahap Perancangan	30
3.2.1	Perancangan Sistem	30
3.2.2	Perancangan Data.....	32
3.2.3	Perancangan Proses.....	34
BAB IV IMPLEMENTASI.....		47
4.1	Lingkungan Implementasi	47
4.1.1	Perangkat Keras	47
4.1.2	Perangkat Lunak	47
4.2	Implementasi Tahap <i>Preprocessing</i>	47
4.2.1	Normalisasi Intensitas Citra	48
4.2.2	Implementasi <i>Preprocessing</i> ROI atas.....	48
4.2.3	Implementasi <i>Preprocessing</i> ROI bawah.....	48
4.3	Implementasi Proses Ekstraksi Fitur	49
4.3.1	Implementasi Proses Filter Gaussian 2D	49
4.3.2	Implementasi Proses Filter Log Gabor	50
4.3.3	Implementasi Proses <i>Phase Congruency</i>	51
4.3.4	Implementasi Membuat <i>Region</i> hasil <i>Laplacian of Gaussian</i>	58
4.4	Implementasi Proses Segmentasi	60
4.4.1	Pembangunan Data Training.....	60
4.4.2	Proses Klasifikasi <i>training</i> menggunakan <i>Extreme Learning Machine</i>	62
4.4.3	Proses Klasifikasi <i>testing</i> menggunakan <i>Extreme Learning Machine</i>	63
4.5	Implementasi <i>Postprocessing</i>	64

BAB V UJI COBA DAN EVALUASI	67
5.1 Lingkungan Uji Coba.....	67
5.2 Data Uji Coba.....	67
5.3 Uji Coba <i>Preprocessing</i>	69
5.3.1 Uji Coba Penentuan Ukuran Matrik Konvolusi pada Filter Mean dan Filter Median	69
5.3.2 Uji Coba Penentuan Ukuran Matrik Konvolusi pada filter <i>motion blur</i>	70
5.4 Uji Coba Ekstraksi Fitur.....	71
5.4.1 Uji Coba Penentuan Parameter Sigma pada filter Gaussian 2D.....	71
5.4.2 Uji Coba Penentuan Parameter Sigma dan <i>Wavelength</i> pada Metode Log Gabor	72
5.4.3 Uji Coba Penentuan Nilai Skala Wavelet dan <i>Wavelength</i> Awal pada <i>Phase Congruency</i>	73
5.4.4 Uji Coba Penentuan Nilai <i>Threshold</i> Binerisasi Hasil <i>Phase Congruency</i>	75
5.5 Uji Coba Kinerja Sistem Segmentasi	76
5.5.1 ROI atas	77
5.5.2 ROI bawah	82
5.6 Evaluasi	84
BAB VI KESIMPULAN DAN SARAN.....	89
6.1 Kesimpulan	89
6.2 Saran.....	90
DAFTAR PUSTAKA.....	91
BIODATA PENULIS.....	93

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Dental panoramic radiograph.	7
Gambar 2.2 Full mouth series.	9
Gambar 2.3 Focal trough.....	10
Gambar 2.4 Cara pengambilan foto radiograph.	10
Gambar 2.5 Hasil foto fental panoramic radiograph.	10
Gambar 2.6 Citra ROI.	11
Gambar 2.7 Trabecular bone dan cortical bone.....	12
Gambar 2.8 Struktur trabecular bone.	13
Gambar 2.9 <i>Motion Blur</i>	16
Gambar 2.10 Perbedaan domain frekuensi pada filter Gabor dan filter Log Gabor.	18
Gambar 2.11 Fungsi transfer simetris genap	19
Gambar 2.12 Fungsi transfer simetris ganjil	19
Gambar 2.13 Ilustrasi penggabungan fungsi transfer filter simetris ganjil dan genap.....	20
Gambar 2.14 Filter Log Gabor pada domain Fourier.....	20
Gambar 2.15. Perbandingan hasil <i>Phase congruency</i>	22
Gambar 3.1 ROI bawah. Area dalam kotak kuning memperlihatkan rendahnya kontras	30
Gambar 3.2 Detail Struktur <i>Trabecular Bone</i> terdapat diantara dua akar gigi	30
Gambar 3.3 Diagram alir dari sistem aplikasi segmentasi <i>trabecular bone</i>	32
Gambar 3.4 ROI yang digunakan sebagai input.....	33
Gambar 3.5 citra output yang diharapkan.	33
Gambar 3.6 Diagram Alir Tahap <i>Preprocessing</i>	38
Gambar 3.7 Hasil <i>preprocessing</i>	39
Gambar 3.8 Diagram Alir Pengambilan fitur intensitas citra	39
Gambar 3.9 Diagram Alir Filter Gaussian 2D.....	40
Gambar 3.10 Citra Hasil Ekstraksi Fitur Gaussian 2D.....	40
Gambar 3.11 Diagram Alir Ekstraksi Fitur Filter Log Gabor	41
Gambar 3.12 Hasil Akhir Ekstraksi Fitur Filter Log Gabor.....	41
Gambar 3.13 Diagram alir proses ekstraksi fitur dengan <i>phase congruency</i>	42

Gambar 3.14 Hasil Output <i>Phase Congruency</i>	43
Gambar 3.15 Diagram alir proses ekstraksi fitur dengan <i>Laplacian of Gaussian</i>	43
Gambar 3.16 Hasil Output <i>Laplacian of Gaussian</i>	43
Gambar 3.17 Diagram Alir Klasifikasi segmen menggunakan ELM.....	44
Gambar 3.18 Citra hasil segmentasi	44
Gambar 3.19 Diagram alir <i>postprocessing</i>	45
Gambar 3.20 Citra hasil <i>postprocessing</i>	45
Gambar 3.21 Rancangan Antarmuka.....	46
Gambar 5.1 Dental panoramic radiograph.....	68
Gambar 5.2 Citra hasil cropping.....	68
Gambar 5.3 Citra groundtruth.	69
Gambar 5.4 Hasil uji coba filter Mean dan filter Median.....	70
Gambar 5.5 Hasil uji coba filter <i>Motion Blur</i>	71
Gambar 5.6 Hasil uji coba filter Gaussian 2D.....	72
Gambar 5.7 Matrik korelasi antara hasil filter Gaussian 2D	72
Gambar 5.8 Hasil uji coba filter Log Gabor.....	73
Gambar 5.9 Hasil uji coba <i>phase congruency</i>	74
Gambar 5.10 Hasil uji coba binerisasi <i>phase congruency</i>	75
Gambar 5.11 Citra dengan performa sensitivitas tertinggi	79
Gambar 5.12 Citra dengan performa sensitivitas terendah.....	79
Gambar 5.13 Citra dengan performa spesifisitas tertinggi	80
Gambar 5.14 Citra dengan performa spesifisitas dan akurasi terendah.....	80
Gambar 5.15 Citra dengan performa akurasi tertinggi	80
Gambar 5.16 Citra dengan performa sensitivitas tertinggi	82
Gambar 5.17 Citra dengan performa sensitivitas terendah.....	83
Gambar 5.18 Citra dengan performa spesifisitas tertinggi	83
Gambar 5.19 Citra dengan performa spesifisitas terendah dan akurasi terendah	83
Gambar 5.20 Citra dengan performa akurasi tertinggi	84
Gambar 5.21 Citra cortical bone yang terputus pada hasil <i>phase congruency</i>	86
Gambar 5.22 Citra cortical bone yang terputus pada <i>LoG</i>	87

DAFTAR TABEL

Tabel 2.1. Pembanding hasil segmentasi dengan ground truth ...	25
Tabel 2.2 Metode pengukuran performa segmentasi	26
Tabel 5.1 Hasil Confusion Matrix	76
Tabel 5.2 Hasil pengujian segmentasi ROI atas	78
Tabel 5.3 Hasil pengujian segmentasi ROI bawah.....	81

[Halaman ini sengaja dikosongkan]

KODE SUMBER

Kode Sumber 4.1 Implementasi Fungsi Normalisasi Citra	48
Kode Sumber 4.2 Implementasi fungsi <i>preprocessing</i> ROI atas.	48
Kode Sumber 4.3 Implementasi fungsi <i>preprocessing</i> ROI bawah	49
Kode Sumber 4.4 Implementasi filter Gaussian 2D.	49
Kode Sumber 4.5 Implementasi fungsi proses filter Log Gabor.	50
Kode Sumber 4.6 Implementasi <i>phase congruency</i>	55
Kode Sumber 4.7 Implementasi mendapatkan <i>region</i> menggunakan <i>phase congruency</i>	58
Kode Sumber 4.8 Implementasi mendapatkan <i>region</i> menggunakan <i>Laplacian of Gaussian</i>	60
Kode Sumber 4.9 Implementasi pembangunan data training.	62
Kode Sumber 4.10 Implementasi proses klasifikasi <i>training</i> menggunakan <i>Extreme Learning Machine</i>	63
Kode Sumber 4.11 Implementasi proses klasifikasi <i>testing</i> menggunakan <i>Extreme Learning Machine</i>	64
Kode Sumber 4.12 Implementasi proses <i>postprocessing</i>	65

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini dibahas hal-hal yang mendasari Tugas Akhir. Bahasan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan Tugas Akhir.

1.1 Latar Belakang

Dental panoramic radiograph adalah citra x-ray dua dimensi (2-D) dari gigi yang merekam keseluruhan mulut, termasuk gigi, rahang atas, rahang bawah dan jaringan serta struktur yang melingkupinya dalam satu citra. *Dental panoramic radiograph* adalah citra yang mudah dan murah diperoleh sehingga seringkali dipergunakan sebagai alat bantu diagnosa penyakit-penyakit serius seperti osteoporosis [1]. Pada dental panoramic radiograph terekam berbagai informasi yang dapat digunakan sebagai indikator dalam mengidentifikasi berbagai penyakit diantaranya struktur *trabecular bone*. Namun, pada dental panoramic radiograph, *trabecular bone* cukup sulit untuk diamati karena kualitas citra yang buruk, rendahnya kontras citra, pencahayaan yang tidak merata serta banyaknya *noise*.

Penelitian sebelumnya yang memanfaatkan *dental panoramic radiographs* untuk mendeteksi *trabecular bone* telah dilakukan oleh Dewi [2]. Penelitian ini menggunakan struktur linear untuk mendeteksi daerah *trabecular bone*. Metode yang digunakan adalah menggunakan *multiscale linear operator* dan *clustering* untuk membedakan gigi dan *trabecular bone*. Penelitian lain yang memanfaatkan *dental panoramic radiographs* untuk mendeteksi *trabecular bone* dilakukan oleh Yuniarti [3]. Penelitian ini menggunakan informasi tekstural menggunakan pengukuran statistik yaitu *first and second order measure*. Penelitian ini dilakukan pada ROI rahang bawah untuk membedakan daerah *trabecular bone* dan *cortical bone*.

ELM merupakan jaringan syaraf tiruan *feedforward* dengan satu *hidden layer* atau lebih dikenal dengan istilah *single hidden layer feedforward neural network* (SLFNs). Metode ELM mempunyai kelebihan dalam kompleksitas *learning* dikarenakan metode ini

hanya melakukan satu kali proses *training weight*. Penelitian yang dilakukan oleh Zhu dkk. [4] menggunakan ELM membuktikan bahwa ELM memiliki waktu training dan waktu segmentasi yang lebih cepat daripada SVM, *Random Forest*, dan *AdaBoost* dalam segmentasi pembuluh darah retina.

Tugas akhir ini mengusulkan segmentasi area *trabecular bone* pada *dental panoramic radiograph* berbasis karakteristik profil segmen dan metode klasifikasi *Extreme Learning Machine* (ELM). Profil karakteristik segmen dari citra ini dipilih untuk melakukan ekstraksi fitur pada proses segmentasi. Profil karakteristik yang dimaksud adalah mengambil nilai intensitas yang dihasilkan oleh berbagai filter.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana melakukan *preprocessing* citra *area trabecular bone dental panoramic*?
2. Bagaimana melakukan ekstraksi fitur pada *dental panoramic radiograph*?
3. Bagaimana mengklasifikasi piksel dari data fitur sebagai *area trabecular bone* dan *area* lain menggunakan *Extreme Learning Machine*?

1.3 Batasan Masalah

Batasan masalah yang terdapat pada Tugas Akhir ini, yaitu sebagai berikut :

1. Citra yang digunakan adalah *dental panoramic radiograph* bertipe *grayscale* (keabuan).
2. *Region of interest* yang diambil dari masing-masing *dental panoramic radiograph* pada *region trabecular bone* yang mengandung gigi dan *cortical bone* masing-masing pada bagian

kanan dan kiri pada daerah rahang bawah dengan ukuran 256x128 piksel.

3. Metode ini diimplementasikan menggunakan MATLAB 2014a.

1.4 Tujuan Tugas Akhir

Tujuan tugas akhir ini mengusulkan segmentasi area *trabecular bone* pada *dental panoramic radiograph* berbasis karakteristik profil segmen dan metode klasifikasi *Extreme Learning Machine*.

1.5 Manfaat Tugas Akhir

Manfaat dari tugas akhir ini diharapkan dapat menghasilkan sistem untuk mensegmentasi *trabecular bone* pada *dental panoramic radiograph* dengan performa yang baik. Hasil segmentasi area *trabecular bone* dapat dijadikan dasar dalam perhitungan informasi struktur *trabecular bone*. Informasi struktur *trabecular bone* yang dapat dihitung diantaranya percabangan, luas rongga antartulang dan sebagainya. Informasi-informasi tersebut dapat digunakan dalam identifikasi contohnya penyakit osteoporosis. Oleh karena itu dibutuhkan hasil segmentasi yang akurat sehingga dapat memberikan hasil perhitungan informasi struktur *trabecular bone* yang lebih akurat pula.

1.6 Metodologi

Metodologi yang digunakan pada pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Studi Literatur

Pada tahap ini dilakukan pencarian literatur berupa jurnal atau *paper* yang digunakan sebagai referensi untuk pengerjaan tugas akhir ini. Literatur yang digunakan terutama pada tahap ekstraksi fitur hingga klasifikasi menggunakan ELM adalah metode yang diusulkan oleh Zhu, C [4] pada jurnal dari penelitian yang berjudul “*Retinal vessel segmentation in colour fundus images using Extreme Learning Machine*”. Selain itu, referensi penunjang yang akan digunakan adalah

jurnal yang membahas tentang *pre processing* dan ekstraksi fitur yang digunakan pada tugas akhir ini seperti, filter mean, filter median, filter *motion blur*, filter Gaussian 2D, filter Log Gabor, *phase congruency*.

2. Analisis dan desain perangkat lunak
Sebelum implementasi, dilakukan analisis dan perancangan sistem. Analisis dan perancangan sistem diperlukan agar implementasi berjalan secara terstruktur dan memberikan hasil sesuai kebutuhan.

Aplikasi dilengkapi dengan UI yang akan menampilkan citra asli yang dilengkapi kotak yang menunjukkan ROI, citra hasil *cropping* ROI bagian kiri dan kanan dan citra hasil segmentasi *region trabecular bone*. Input aplikasi berupa citra *dental panoramic radiograph* berukuran 1976 x 976 piksel. Output aplikasi berupa hasil segmentasi *trabecular bone* pada citra ROI yang dipilih.

3. Implementasi Algoritma
Proses pertama adalah pemilihan *region of interest*, dilakukan secara manual dengan memilih empat area persegi berbeda yaitu dua kanan dan dua kiri dengan ukuran 128x128 piksel pada tulang rahang bawah pada area *trabecular bone* yang mengandung area gigi untuk ROI pertama dan *cortical bone* pada ROI kedua. Proses berikutnya adalah melakukan *preprocessing*. Citra tersebut diekstraksi menjadi beberapa fitur pada setiap pikselnya. Kemudian hasil ekstraksi fitur ini digunakan pada proses segmentasi menggunakan metode klasifikasi *Extreme Learning Machine*. Kemudian hasil klasifikasi tersebut dilakukan *postprocessing* untuk membersihkan citra.

4. Pengujian dan Evaluasi
Dalam tahap ini sistem diuji dengan citra *dental panoramic radiograph* dari data testing, kemudian membandingkan luarannya dengan *groundtruth*. Performa dari sistem dievaluasi dengan melihat nilai akurasi, *sensitivity*, *specificity*, dan waktu eksekusi yang dibutuhkan.

1.7 Sistematika Laporan

Buku Tugas Akhir ini disusun dengan sistematika sebagai berikut:

1. Bab I Pendahuluan
Pada bab ini dibahas latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, dan sistematika laporan Tugas Akhir.
2. Bab II Tinjauan Pustaka
Pada bab ini dibahas dasar teori yang berkaitan dengan topik Tugas Akhir.
3. Bab III Perancangan
Pada bab ini dibahas perancangan metode-metode baik ekstraksi fitur dan *classifier* yang digunakan.
4. Bab IV Implementasi
Pada bab ini dibahas implementasi rancangan yang dibuat pada tahap perancangan.
5. Bab V Uji Coba dan Evaluasi
Pada bab ini dibahas tahap-tahap uji coba. Kemudian hasil uji coba dievaluasi untuk mengetahui kinerja dari aplikasi yang dibangun.
6. Bab VI Kesimpulan dan Saran
Pada bab ini disimpulkan hasil perbandingan. Dibahas juga saran untuk pengembangan lebih lanjut.

[Halaman ini sengaja dikosongkan]

BAB II DASAR TEORI

Pada bab ini akan diuraikan mengenai dasar-dasar teori yang digunakan dalam pengerjaan tugas akhir ini. Dalam bab ini akan dituliskan penjelasan *dental panoramic radiograph*, *trabecular bone* dan *Region of Interest (ROI)* sebagai ruang lingkup dan studi kasus. Kemudian dilanjutkan *preprocessing* yang digunakan seperti normalisasi, filter mean, filter median, dan *motion blur filter*. Lalu metode ekstraksi fitur yang digunakan seperti filter Gaussian 2D, filter log Gabor, dan *phase congruency* dalam proses ekstraksi fitur, metode *Extreme Learning Machine* dalam proses segmentasi.

2.1 Dental Panoramic Radiograph dan Struktur Tulang

Dental panoramic radiograph merupakan foto *radiograph* pada bagian rahang manusia. Contoh *dental panoramic radiograph* ditunjukkan pada Gambar 2.1. Dengan melihat gambar tersebut, terlihat susunan tulang dan gigi pada rahang manusia atau mungkin bahan-bahan tambahan lainnya seperti gigi palsu atau aksesoris gigi yang bermacam-macam jenisnya.



Gambar 2.1 Dental panoramic radiograph.

2.1.1 Pengertian Dental Radiograph

Radiograph adalah sebuah citra yang dihasilkan pada sebuah *film* fotografik spesial atau piringan dengan menggunakan sinar radiasi. Tipe radiasi *elektromagnetic* yang paling menarik adalah sinar *X* dan radiasi *gamma*. Radiasi tersebut lebih *energetic* daripada *radio waves* dan *visible light*. Sinar *gamma* memiliki energi yang tinggi, oleh karena itu tidak hanya berguna dalam *radiography* tetapi juga berbahaya bagi kehidupan organisme. [5]

Radiography banyak digunakan pada bidang industri dan bidang kesehatan (medis). *Radiography* pada bidang kesehatan menggunakan radiasi *elektromagnetic X-Rays* untuk melihat benda yang tak terlihat atau benda *Hard-to-Image*. Tujuan utama *radiography* dengan menggunakan sinar *X-Rays* adalah untuk melihat isi dalam tubuh manusia di mana tulang sebagai benda keras dapat ditampilkan dengan resolusi optimum (128 tingkat keabu-abuan) seperti halnya bagian otak, *dental* dan *fetus*. [5]

Dental X-Rays merupakan citra *radiograph* dari gigi, tulang, dan jaringan lembut di sekitarnya. *Dental X-Rays* digunakan untuk mengidentifikasi permasalahan yang menyangkut gigi, mulut, dan rahang. Citra *X-Ray* dapat memperlihatkan lubang pada gigi, susunan gigi yang tersembunyi, dan tulang yang seolah-olah menghilang yang tidak bisa dilihat pada saat pemeriksaan visual. *Dental X-Rays* juga dapat dapat dilakukan sebagai kelanjutan dan pengecekan hasil dari perawatan gigi. [6]

Citra *radiography* dibuat dengan mengontrol penuh sinar radiasi *X-Ray* dengan tingkatan yang berbeda-beda tergantung pada anatomi gigi. Pada citra hasil radiasi, gigi terlihat lebih terang karena gigi merupakan benda yang keras sehingga hanya sedikit sinar radiasi yang dapat menembusnya. Sebaliknya, kerusakan gigi, gigi yang berlubang, infeksi gigi, dan adanya perubahan pada struktur tulang akan terlihat lebih gelap dikarenakan sinar radiasi lebih mudah menembusnya. Untuk gigi yang telah mengalami perbaikan, misalnya ditambal atau untuk gigi yang diberi aksesoris seperti emas atau perak,

penampakan yang terlihat bergantung pada jenis bahan tambahan tersebut, bisa terlihat lebih terang atau lebih gelap serti pada Gambar 2.2. [6]

2.1.2 Teori Panoramic Radiographs

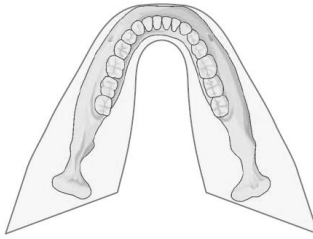
Panaromic radiography adalah tipe modifikasi dari *tomography* atau citra layer *radiography*. Pada *panaromic radiography*, bagian rahang (*dental*) pasien yang melengkung harus diposisikan didalam *narrow zone of sharp focus* yang dikenal sebagai layer citra atau "*focal trough*" seperti yang ditunjukkan pada Gambar 2.3. [7]

Gigi dan struktur yang terdapat pada luar zona *sharp focus* akan terlihat buram, akibat adanya *distorsi* atau benda-benda tambahan lainnya. Oleh karena itu, pada semua mesin *panaromic* mempunyai beberapa mekanisme untuk beberapa posisi berbeda sesuai dengan *patient's denition* di dalam *focal trough*. [7] Gambar pada Gambar 2.4 merupakan petunjuk pabrik untuk letak posisi pasien yang sesuai adalah kritis dalam memperoleh sebuah *radiograph* yang berkualitas. Sedangkan Gambar 2.5 adalah foto hasil *radiograph* tampak samping.

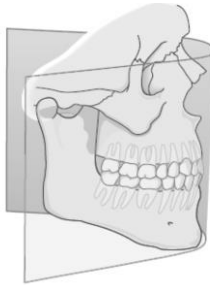
Pembesaran dan *X-ray tube focal spot size* merupakan 2 faktor penting dalam penetapan kualitas citra *extraoral*. Kemampuan sistem citra menghasilkan citra terang dari objek ruang secara dekat atau biasa disebut dengan resolusi adalah sebuah pengukuran objektif kualitas citra dan diekspresikan dalam unit *Line Pair* per milimeter (LP/mm). Seiring dengan adanya perkembangan pada teori resolusi maka kemampuan sistem untuk menghasilkan gambar yang lebih detail semakin meningkat pula. [7]



Gambar 2.2 Full mouth series.



Gambar 2.3 Focal trough.



Gambar 2.4 Cara pengambilan foto radiograph.



Gambar 2.5 Hasil foto fental panoramic radiograph.

2.1.3 Struktur Susunan Dental Panoramic Radiograph

Terdapat banyak jenis dan bentuk objek yang bisa nampak pada *dental panoramic radiograph*, ditunjukkan pada Gambar 2.1. Pemilihan *region of interest* (ROI), dilakukan secara manual dengan memilih dua area persegi berbeda yaitu kanan dan kiri dengan ukuran 256x128 piksel pada tulang rahang bawah pada area *trabecular bone* yang mengandung area gigi namun berada di daerah gusi untuk ROI pertama dan *cortical bone* pada ROI kedua seperti yang ditunjukkan pada Gambar 2.6.

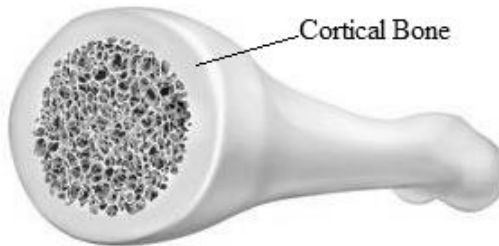


Gambar 2.6 Citra ROI.

2.1.4 Struktur Tulang

Tulang atau rangka adalah penopang tubuh, tanpa tulang pasti tubuh tidak bisa tegak berdiri. Tulang mulai terbentuk sejak bayi dalam kandungan dan terus berlangsung sampai dekade kedua dalam susunan yang teratur. Pertumbuhan tulang selengkapnya terbentuk pada umur lebih kurang 30 tahun. Setelah itu ada juga perubahan yang disebut *remodelling*. Tulang merupakan *reservoir* terbesar dari kalsium dan fosfat. 99% kalsium terdapat ditulang (1000 gram) dari jumlah kalsium tubuh, sedangkan fosfat dalam tulang mencapai 90% dari fosfat dalam tubuh. Setiap tulang yang terdapat pada tubuh

manusia terdiri dari dua bagian yaitu *cortical bone* dan *trabecular bone*. [8]



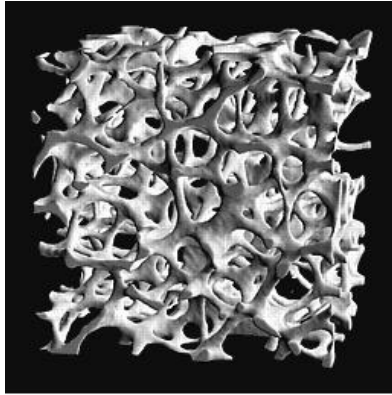
Gambar 2.7 Trabecular bone dan cortical bone.

2.1.5 Cortical bone

Cortical bone atau bisa disebut dengan *compact bone* merupakan bagian tulang yang padat dan kuat, pada Gambar 2.7 ditunjukkan bahwa *cortical bone* terletak pada bagian luar struktur tulang dan memberikan kontribusi sebesar 80% dari total berat tulang manusia. *Cortical bone* adalah lapisan-lapisan tulang yang tersusun, memiliki sedikit rongga, dan mempunyai tekstur yang sangat halus tetapi kuat. *Cortical bone* mengandung banyak kapur (*Calcium Phosphat* dan *Calcium Carbonat*) sehingga menjadi padat dan kuat serta mempunyai fungsi untuk melindungi organ, menyediakan tingkatan pergerakan, dan menyimpan mineral. [8]

2.1.6 Trabecular bone

Trabecular bone atau *cancellous bone* atau *spongy bone* adalah bagian tulang yang berongga seperti yang terlihat pada Gambar 2.8, sangat rapuh, dan mudah keropos tetapi memiliki area permukaan yang sangat luas. Bagian tulang ini terletak dibagian dalam yang dikelilingi oleh *cortical bone*. Pada *trabecular bone* terdapat *bone marrow* yang berfungsi memproduksi komponen sel darah yang biasa disebut dengan *hematopoiesis*. Pada *trabecular bone* juga banyak ditemukan pembuluh darah arteri dan vena. [8]



Gambar 2.8 Struktur trabecular bone.

2.2 Filter

2.2.1 Normalisasi

Normalisasi adalah teknik *preprocessing* untuk melakukan penskalaan dan penstandaran data. Pada tugas akhir ini, normalisasi dilakukan dua kali, yang pertama adalah normalisasi citra dan yang kedua adalah normalisasi vektor fitur. Normalisasi kedua data tersebut menggunakan metode normalisasi skala.

2.2.1.1 Normalisasi Skala

Normalisasi skala yaitu melakukan penskalaan data pada rentang tertentu, di mana rentang yang umum digunakan adalah rentang 0-1. Normalisasi vektor fitur pada tugas akhir ini menggunakan normalisasi skala dengan tujuan untuk menghindari rentang data yang terlalu jauh. Formulasi untuk menghitung normalisasi skala ditunjukkan pada Persamaan 2.1, di mana X_{max} dan X_{min} adalah nilai tertinggi dan terendah dari satu fitur yang sama. Nilai fitur setelah normalisasi disimbolkan dengan Y , sedangkan X adalah nilai fitur sebelum normalisasi.

$$Y = \frac{X - X_{min}}{X_{max} - X_{min}}. \quad (2.1)$$

2.2.1.2 Normalisasi Skala Rentang Baru

Normalisasi skala rentang baru hampir sama dengan normalisasi skala 0-1. Perbedaan keduanya hanya pada rentang yang dijadikan acuan. Normalisasi ini digunakan untuk standardisasi citra supaya semua citra memiliki rentang keabuan yang sama. Formulasi normalisasi tersebut ditunjukkan pada Persamaan 2.2. Salah satu citra yang dipilih sebagai acuan dihitung terlebih dahulu rentang minimum dan maksimum intensitasnya, yang disimbolkan dengan $X_{2\min}$ dan $X_{2\max}$. Rentang keabuan terendah dan tertinggi pada citra awal dilambangkan dengan $X_{1\min}$ dan $X_{1\max}$, sedangkan intensitas keabuan sebelum normalisasi dan setelah normalisasi dilambangkan dengan X dan Y .

$$Y = \frac{X - X_{1\min}}{X_{1\max} - X_{1\min}} \times (X_{2\max} - X_{2\min}). \quad (2.2)$$

2.2.2 Filter Median

Metode median filter merupakan filter non-linear yang dikembangkan Tukey, yang berfungsi untuk menghaluskan dan mengurangi *noise* atau gangguan pada citra. Dikatakan nonlinear karena cara kerja penapis ini tidak termasuk kedalam kategori operasi konvolusi. Operasi nonlinear dihitung dengan mengurutkan nilai intensitas sekelompok pixel, kemudian menggantikan nilai pixel yang diproses dengan nilai tertentu.

Pada median filter suatu *window* atau penapis yang memuat sejumlah pixel ganjil digeser titik per titik pada seluruh daerah citra. Nilai-nilai yang berada pada window diurutkan secara ascending untuk kemudian dihitung nilai mediannya. Nilai tersebut akan menggantikan nilai yang berada pada pusat bidang *window*.

Jika suatu *window* ditempatkan pada suatu bidang citra, maka nilai pixel pada pusat bidang window dapat dihitung dengan mencari nilai median dari nilai intensitas sekelompok piksel yang telah diurutkan. Secara matematis dapat dirumuskan pada Persamaan 2.3 dimana $g(x,y)$ merupakan citra yang dihasilkan dari citra $f(x,y)$ dengan

w sebagai *window* yang ditempatkan pada bidang citra dan (i,j) elemen dari *window* tersebut. [9]

$$g(x, y) = \text{Median}\{f(x-i, y-j), (i, j) \in w\} \quad (2.3)$$

2.2.3 Filter Mean

Metode mean filter adalah satu teknik filtering yang bekerja dengan cara menggantikan intensitas suatu pixel dengan rata-rata nilai pixel dari pixel-pixel tetangganya. Jika suatu citra $f(x,y)$ yang berukuran $M \times N$ dilakukan proses filtering dengan penapis $h(x,y)$ maka akan menghasilkan citra $g(x,y)$, dimana penapis $h(x,y)$ merupakan matrik yang berisi nilai 1/ukuran penapis. Secara matematis proses tersebut dapat dinyatakan pada Persamaan 2.4.

$$g(x,y) = f(x,y) * h(x,y) \quad (2.4)$$

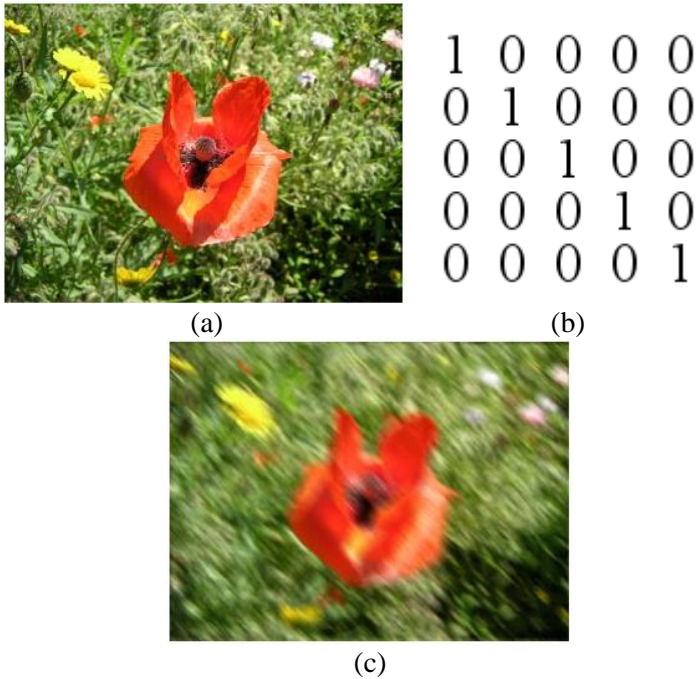
Operasi tersebut dipandang sebagai konvolusi antara citra $f(x,y)$ dengan penapis $h(x,y)$, dimana $*$ menyatakan operator konvolusi dan prosesnya dilakukan dengan menggeser penapis konvolusi pixel per pixel. [9]

2.2.4 Filter Motion Blur

Motion blur adalah objek melesat yang bergerak cepat dalam gambar diam atau urutan gambar seperti film atau animasi. Ini terjadi ketika gambar sedang direkam perubahan selama perekaman dari paparan tunggal, baik karena gerakan cepat atau paparan panjang. *Motion blur* didapatkan dengan mem-*blur* pada satu arah saja. Contoh penggunaan dan matriks yang digunakan pada motion blur dipaparkan dalam Gambar 2.9. [10]

2.3 Ekstraksi fitur

Dalam sub bab ini dituliskan penjelasan metode ekstraksi fitur yang digunakan dalam tugas akhir ini. Ada 3 metode utama, yaitu Filter *Gaussian* 2D, filter Log Gabor, *Phase Congruency*, dan *Laplacian of Gaussian (LoG)*.



Gambar 2.9 Motion Blur

(a) citra asli (b) matriks filter *motion blur* (c) hasil *motion blur*

2.3.1 Filter Gaussian 2D

Filter *Gaussian* adalah filter yang menggunakan fungsi *Gaussian*. *Gaussian filtering* didapat dari operasi konvolusi. Operasi perkalian yang dilakukan ialah perkalian antara matriks kernel dengan matriks gambar asli. Matriks kernel gauss didapat dari fungsi komputasi dari distribusi *Gaussian*, seperti pada Persamaan 2.5 dimana x dan y adalah posisi sumbu- x dan sumbu- y berdasarkan pusat. Parameter sigma ini berpengaruh pada filter 2D-*Gaussian* ini [11].

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (2.5)$$

2.3.2 Filter Log Gabor

Filter *Gabor* adalah pilihan tradisional untuk memperoleh informasi frekuensi lokal. Filter ini menawarkan lokalisasi simultan terbaik dari informasi spasial dan frekuensi. Namun filter ini memiliki dua keterbatasan utama. *Bandwidth* maksimum dari filter *Gabor* terbatas pada sekitar satu oktaf dan filter *Gabor* tidak optimal dalam mencari informasi spektral yang luas dengan lokalisasi spasial yang maksimal.

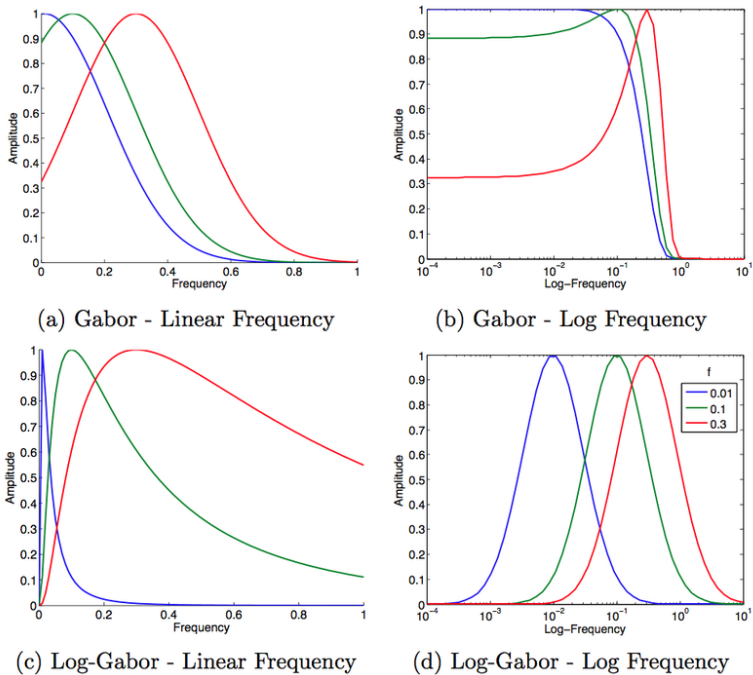
Sebuah alternatif untuk fungsi *Gabor* adalah fungsi *Log-Gabor* diusulkan oleh Field (1987) [12]. Filter *Log-Gabor* dapat dibangun dengan *bandwidth* yang berubah-ubah dan *bandwidth* dapat dioptimalkan untuk menghasilkan filter dengan batas spasial.

Field menunjukkan bahwa citra asli yang lebih baik dikodekan dengan filter yang memiliki fungsi mentransfer *Gaussian* bila dilihat dari skala frekuensi logaritmik. Fungsi *Gabor* memiliki fungsi mentransfer *Gaussian* bila dilihat dari skala frekuensi linear seperti yang ditunjukkan pada Gambar 2.10. Pada skala frekuensi linear fungsi *Log-Gabor* memiliki fungsi transfer dari persamaan (2.6) dimana w_0 adalah frekuensi pusat filter. Untuk mendapatkan rasio bentuk konstan $\frac{k}{w_0}$ juga harus tetap konstan untuk berbagai w_0 . Misalnya, $\frac{ka}{w_0}$ nilai 0,74 akan menghasilkan *bandwidth* filter sekitar satu oktaf, 0,55 akan menghasilkan dua oktaf, dan 0,41 akan menghasilkan tiga oktaf.

$$G(w) = e^{(-\log(w/w_0)^2) / (2 (\log(k/w_0))^2)} \quad (2.6)$$

Ada dua karakteristik penting untuk diperhatikan. Pertama, fungsi *Log-Gabor* dipastikan tidak memiliki komponen DC. Kedua, fungsi transfer pada fungsi *Log Gabor* memiliki ekor yang diperpanjang pada akhir frekuensi tinggi. Studi Field tentang statistik dari suatu gambar menunjukkan bahwa suatu gambar memiliki

spektrum amplitudo yang jatuh pada sekitar $\frac{1}{w}$. Untuk mengkodekan suatu gambar yang memiliki karakteristik spektral, diharuskan menggunakan filter yang memiliki spektrum yang mirip.



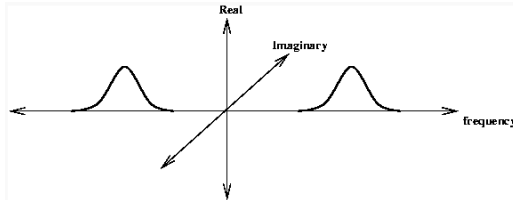
Gambar 2.10 Perbedaan domain frekuensi pada filter Gabor dan filter Log Gabor.

Field menunjukkan bahwa fungsi *Log Gabor* dapat mengkodekan suatu gambar lebih efisien daripada fungsi *Gabor* biasa, yang terlalu berlebihan merepresentasikan komponen frekuensi rendah dan kurang dalam merepresentasikan komponen frekuensi tinggi dalam pengkodean apapun. Hal lain yang mendukung fungsi *Log Gabor* adalah bahwa filter ini konsisten dengan pengukuran pada

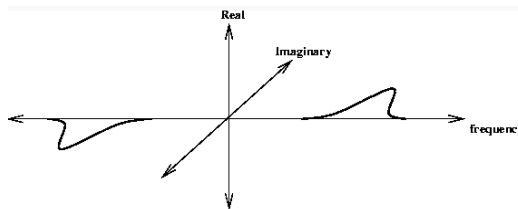
sistem visual manusia yang menunjukkan bahwa manusia memiliki respon sel yang simetris pada skala frekuensi log.

Dalam domain frekuensi, filter simetris genap diwakili oleh dua bernilai real puncak *Log-Gaussian* simetris ditempatkan pada setiap sisi dari titik asal seperti yang ditunjukkan pada Gambar 2.11. Filter simetris ganjil, seperti yang ditunjukkan pada Gambar 2.12, direpresentasikan oleh dua nilai imajiner puncak *Log-Gaussian* anti-simetris yang ditempatkan pada setiap sisi titik asal.

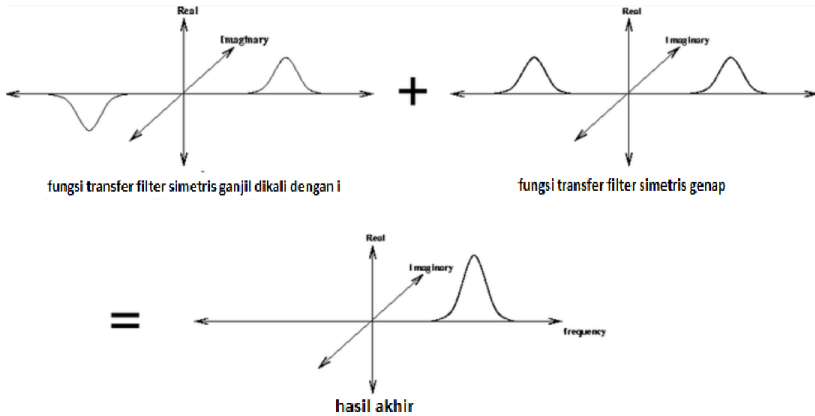
Konvolusi dari filter simetris genap dan ganjil tersebut dapat digabungkan dalam satu operasi. Memanfaatkan linearitas dari *Fourier Transform* dimana $FFT(A + B) = FFT(A) + FFT(B)$ kita dapat melakukan hal berikut: Kalikan FFT dari filter simetris ganjil dengan nilai *imaginer* (i) agar hasilnya menjadi real dan ditambahkan dengan FFT filter genap bahkan simetris. Puncak anti-simetris dari filter simetris ganjil akan menghilangkan puncak simetris dari filter simetris genap. Hal ini membuat satu puncak (dikalikan dengan 2) berpindah ke sisi positif dari spektrum frekuensi seperti diilustrasikan pada Gambar 2.13.



Gambar 2.11 Fungsi transfer simetris genap

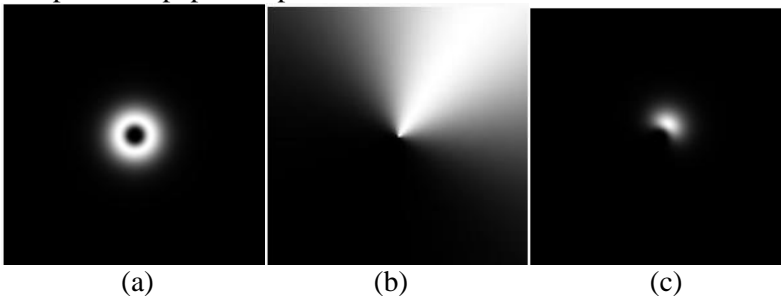


Gambar 2.12 Fungsi transfer simetris ganjil



Gambar 2.13 Ilustrasi penggabungan fungsi transfer filter simetris ganjil dan genap

Filter yang dibangun pada *Log-Gabor* ini dapat menggunakan dua komponen: komponen *radial*, yang mengontrol pita frekuensi pada respon filter dan komponen *angular*, yang mengontrol orientasi pada respon filter. Filter yang digunakan pada masing-masing komponen di paparkan pada Gambar 2.14.



Gambar 2.14 Filter Log Gabor pada domain Fourier

(a) Filter Radial log-Gabor (b) komponen filter angular (c) filter Angular Log-Gabor

2.3.3 Phase Congruency

Phase congruency menunjukkan perubahan kontras dan kecerahan citra yang invarian. Untuk mengomputasi seluruh pengukuran dalam *phase congruency* pada citra, sebuah model efisien diusulkan oleh Kovessi [13] pada Persamaan 2.7 dimana (x, y) adalah posisi dari piksel. $A_{k,\theta}$ adalah citra yang telah difilter menggunakan filter *Log Gabor*. k adalah skala, and θ adalah enam arah dari 0 sampai π . $\lfloor \rfloor$ menunjukkan bahwa nilai terdekat sama dengan nilai dirinya sendiri bila positif, dan nol untuk selain bilangan positif. ε adalah konstan yang bernilai kecil dan digunakan untuk mencegah pembagian nol dimana energi lokal sangat kecil pada citra jaringan.

$$PC(x, y) = \frac{\sum_{\theta} \sum_k W_{\theta}(x, y) [A_{k,\theta} \Delta \Phi_{k,\theta}(x, y) - T_{\theta}]}{\sum_{\theta} \sum_k A_{k,\theta}(x, y) + \varepsilon} \quad (2.7)$$

T_{θ} adalah respons *noise* yang digunakan untuk mengurangi *noise* pada hasil perhitungan energi. *Response noise* tersebut dapat dihitung menggunakan Persamaan 2.8 dimana τ adalah hasil akar dari estimasi energi *noise* dan ρ adalah estimasi efek *noise* diset 1.7.

$$T_{\theta} = \frac{\tau \sqrt{\frac{\pi}{2}} + k * \sqrt{\left(2 - \frac{\pi}{2}\right) \tau^2}}{\rho} \quad (2.8)$$

W_{θ} adalah faktor pemberat yang mendevaluasi *Phase Congruency* pada lokasi dimana sebaran filter sangat tipis. W_{θ} didapatkan dengan fungsi *sigmoid* pada persamaan (2.9) dimana c adalah nilai pembatas dari respons filter yang menyebar. γ adalah faktor pembangkit yang mengontrol ketajaman pembatas.

$$W(x, y) = \frac{1}{1 + e^{\gamma(c - s(x, y))}} \quad (2.9)$$

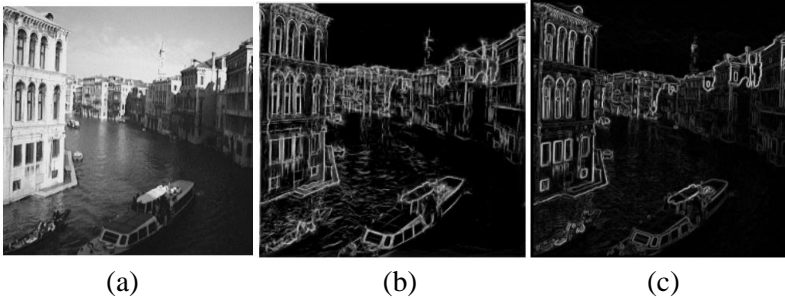
$S(X)$ didapat dengan menjumlahkan amplitudo respons filter dan dibagi dengan amplitudo maksimum pada tiap titik pada citra. Rumus

mencari $S(X)$ didefinisikan pada Persamaan 2.10 dimana K adalah jumlah total skala, ε digunakan untuk menghindari pembagian nol. Nilai $S(X)$ ini memiliki rentang antara 0 dan 1.

$$s(x, y) = \frac{1}{K} \left(\frac{\sum_k A_k(x, y)}{A_{\max}(x, y) + \varepsilon} \right) \quad (2.10)$$

Phase Congruency mencerminkan perilaku gambar dalam domain frekuensi. Telah dicatat bahwa fitur tepi memiliki banyak komponen frekuensi dalam fase yang sama. Konsep ini mirip dengan koherensi, kecuali bahwa itu berlaku untuk fungsi panjang gelombang yang berbeda.

Contoh pada dekomposisi *Fourier* dari gelombang persegi terdiri dari fungsi sinus, dimana frekuensi merupakan kelipatan ganjil dari frekuensi dasar. Pada meningkatnya tepi gelombang persegi, masing-masing komponen sinusoidal memiliki fase naik; fase memiliki kongruensi maksimal di tepi. Hal ini terkait dengan tepi yang dipandang oleh manusia dalam sebuah gambar di mana ada perubahan yang tajam antara terang dan gelap. Perbandingan antara hasil *phase congruency* dan *Canny* dipaparkan pada Gambar 2.15.



Gambar 2.15. Perbandingan hasil *Phase congruency*
(a) Citra asli, (b) citra hasil *phase congruency*, (c) citra hasil
canny (sigma=1).

2.3.4 *Laplacian of Gaussian (LoG)*

Laplacian merupakan filter turunan yang fungsinya dapat mendeteksi area yang memiliki perubahan cepat (rapid changes) seperti tepi (edge) pada citra. Namun, laplacian ini sangat rentan atau sensitif terhadap kehadiran derau. Untuk itu, citra yang akan dideteksi tepinya perlu dihaluskan terlebih dahulu dengan menggunakan Gaussian. Dengan demikian dikenal adanya fungsi turunan baru yakni *LoG* atau *Laplacian of Gaussian*. *LoG* memiliki bentuk seperti pada persamaan 2.11.

$$L(x, y) = \nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \quad (2.11)$$

Dengan menggabungkan rumus *Gaussian* pada persamaan 2.5 sebagai $f(x, y)$, didapatkan rumus dari *LoG* seperti yang dipaparkan pada persamaan 2.12 dimana x dan y adalah posisi dihitung dari pusat. [14]

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (2.12)$$

2.4 Metode Klasifikasi *Supervised: Extreme Learning Machine (ELM)*

Supervised classification dapat diartikan sebagai teknik klasifikasi yang memiliki label. Proses klasifikasi didahului dengan identifikasi objek pada data yang menjadi *training*. Data *training* merupakan data yang sudah diklasifikasi atau dikelompokkan sehingga *classifier* akan mendapatkan informasi terlebih dahulu dari data tersebut. Teknik ini berbeda dengan *unsupervised classification*, di mana pengelompokan didasarkan pada karakteristik umum tanpa menyediakan contoh-contoh kelas terlebih dahulu. Beberapa jenis algoritma yang sering digunakan pada *supervised classification* antara lain *decision tree*, *artificial neural network*, *support vector machine (SVM)*, *nearest neighbour*, dan *Extreme Learning Machine*.

Extreme Learning Machine (ELM) yang diusulkan oleh Huang dkk. adalah salah satu *feedforward neural network* yang paling sederhana dengan menggunakan satu *layer hidden node*, dimana *weight* dari *input-hidden* terpilih secara acak dan tidak perlu disesuaikan, dan *weight* dari *hidden-out* didapat menggunakan regresi *least squares* atau regresi *ridge*. Karena model ini hanya mempelajari *weight* dari *hidden-output*, proses *training* menjadi lebih cepat [15].

ELM untuk klasifikasi biner dengan n *hidden-node* pertamanya memetakan input x dengan dimensi sebesar p ke vektor fitur $h(x) = [1, h_1(x), h_2(x), \dots, h_n(x)]^T$ dengan dimensi sebesar $(n+1)$, dimana $h_j(x)$ direpresentasikan pada Persamaan 2.13 dan kemudian diambil label kelas terdekat $y \in \{-1, 1\}$ menggunakan *linear combination* dari komponen vektor fitur $h_j(x)$ seperti didefinisikan pada Persamaan 2.14.

$$h_j(x) = g\left(\sum_{k=1}^p \alpha_{jk} X_k + b_j\right) = g(\alpha_j^T x + b_j), j \in [1, n] \quad (2.13)$$

$$\tilde{y} = \sum_{k=1}^N \beta_k h_k(x) + \beta_0 = \mathbf{h}(x)^T \beta \quad (2.14)$$

$\alpha_j \in R^p$ dan $b_j \in R$ adalah parameter dari hidden nodes yang ke- j yang mana diinisialisasikan secara random dan $\beta = [\beta_0, \beta_1, \dots, \beta_n]^T \in R^{n+1}$ adalah *weight* dari *hidden-out*, $g(\cdot)$ adalah fungsi aktivasi *sigmoid* seperti dijelaskan pada Persamaan 2.15:

$$g(z) = \frac{1}{1 + e^{-z}} \quad (2.15)$$

Weight dari *hidden-out* β dapat dipelajari menggunakan regresi linear atau regresi *ridge*. Diberikan himpunan *training set* $\{(\mathbf{x}^{(i)}, y^{(i)}) \mid \mathbf{x}^{(i)} \in \mathbb{R}^p, y^{(i)} \in \{-1, 1\}\}_{i=1}^m$, didefinisikan matriks $\mathbf{H} \in \mathbb{R}^{m \times (n+1)}$ sebagai output *hidden layer* seperti dijelaskan pada Persamaan matriks 2.16, dan vektor label $\mathbf{y} = [y^{(1)}, \dots, y^{(m)}]^T \in \mathbb{R}^m$,

sehingga β dapat diestimasi dengan regresi *least square* dengan Persamaan 2.17 atau regresi *ridge* seperti pada Persamaan 2.18 dimana λ adalah parameter yang diregularisasi. Klasifikasi biner dari input baru x ditunjukkan dengan rumus seperti yang didefinisikan pada Persamaan 2.19.

$$\mathbf{H} = \begin{bmatrix} 1 & h_1(\mathbf{x}^{(1)}) & \dots & h_n(\mathbf{x}^{(1)}) \\ 1 & h_1(\mathbf{x}^{(2)}) & \dots & h_n(\mathbf{x}^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & h_1(\mathbf{x}^{(m)}) & \dots & h_n(\mathbf{x}^{(m)}) \end{bmatrix}, \quad (2.16)$$

$$\beta^* = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y} \quad (2.17)$$

$$\beta^* = \left(\mathbf{H}^T \mathbf{H} + \lambda \begin{bmatrix} 0 & \\ & \mathbf{I}_n \end{bmatrix} \right)^{-1} \mathbf{H}^T \mathbf{y}, \quad (2.18)$$

$$y = \text{sign}(\mathbf{h}(\mathbf{x})^T \beta^*) \quad (2.19)$$

2.5 Penghitungan Performa

Setelah proses segmentasi, citra *trabecular bone* terbagi menjadi dua kategori, *trabecular* dan *non-trabecular*. Hasil segmentasi dibandingkan dengan citra *ground-truth* menggunakan matriks *Confussion* seperti pada Tabel 2.1. Tabel 2.1. Pembanding hasil segmentasi dengan ground truth

Tabel 2.1. Pembanding hasil segmentasi dengan ground truth

	Hasil Segmentasi: Benar	Hasil Segmentasi: Salah
Ground-Truth: Benar	<i>True Positive (TP)</i>	<i>True Negative (TN)</i>
Ground Truth: Salah	<i>False Positive (FP)</i>	<i>False Negative (FN)</i>

True positive (TP) adalah banyaknya *ground truth* benar yang terklasifikasi dengan benar. *False positive* (FP) adalah banyaknya *ground truth* yang salah terklasifikasi benar. *True negative* (TN) adalah banyaknya *ground truth* benar yang terklasifikasi salah. Dan *false negative* (FN) adalah banyaknya *ground truth* salah yang terklasifikasi dengan salah.

Akurasi (Acc), sensitivitas (Sn), dan spesifisitas (Sp) digunakan untuk mengukur performa segmentasi, seperti yang ditunjukkan pada Tabel 2.2. Akurasi digunakan untuk mengukur performa hasil klasifikasi dibandingkan dengan *ground truth*. Sensitivitas digunakan untuk mengukur hasil klasifikasi yang benar dibanding total *gorund truth* yang bernilai benar. Spesifisitas digunakan untuk mengukur hasil klasifikasi yang salah dibanding total *gorund truth* yang bernilai salah.

Tabel 2.2 Metode pengukuran performa segmentasi

Metode pengukuran	Deskripsi
<i>Sensitivitas (Sn)</i>	$TP/(TP+FN)$
<i>Spesifisitas (Sp)</i>	$TN/(TN+FP)$
<i>Akurasi (Acc)</i>	$(TP+TN)/(TP+FP+TN+FN)$

BAB III

ANALISIS DAN PERANCANGAN

Bab analisis dan perancangan berisi analisis kebutuhan dan perancangan dari aplikasi yang akan dibangun. Tahap analisis membahas mengenai analisis kebutuhan yang menjadi dasar dari tahap perancangan.

3.1 Tahap Analisis

Tahap analisis mendefinisikan kebutuhan yang akan dipenuhi dalam pembangunan aplikasi penentuan segmen *trabecular bone*.

3.1.1 Deskripsi Umum

Pada tugas akhir ini dibangun suatu aplikasi untuk segmentasi *trabecular bone* pada *dental panoramic radiograph*. Aplikasi dapat digunakan untuk menunjang pemanfaatan citra *dental panoramic radiograph* di bidang kesehatan. Data yang digunakan adalah data citra *dental panoramic radiograph*. Data keluaran dari aplikasi merupakan citra hitam putih. Data keluaran tersebut akan digunakan sebagai data masukan pada proses selanjutnya.

3.1.2 Arsitektur Perangkat Lunak

Seluruh proses yang terdapat pada aplikasi segmentasi *trabecular bone* dijalankan pada komputer. Aplikasi menerima data masukan berupa citra *dental panoramic radiograph* yang telah dipilih pada ROI. Aplikasi segmentasi *trabecular bone* dibangun menggunakan bahasa pemrograman MATLAB menggunakan IDE MATLAB R2014a. Aplikasi ini akan berjalan pada *platform* desktop dengan sistem operasi Windows.

Setelah menerima data masukan, aplikasi mengolah data masukan dan menampilkan data keluaran. Data keluaran adalah citra biner segmen *trabecular bone*.

3.1.3 Spesifikasi Kebutuhan Perangkat Lunak

Pada aplikasi segmentasi *trabecular bone* dibutuhkan beberapa fungsi yang dapat memenuhi proses bisnis dalam sistem. Fungsi tersebut antara lain:

1. *Preprocessing*

Preprocessing yang dilakukan pertama kali adalah pemilihan ROI. Kemudian dilakukan normalisasi citra, median filter, dan mean filter untuk ROI bawah. Namun pada ROI atas dilakukan normalisasi citra dan *motion blur* terlebih dahulu.

2. Melakukan ekstraksi fitur dari citra

Ekstraksi fitur dari citra dilakukan dengan menggunakan metode filter Gaussian 2D, filter Log-Gabor, *Phase Congruency*, dan *Laplacian of Gaussian*.

3. Melakukan binerisasi citra segmen *trabecular bone*

Extreme Learning Machine digunakan untuk mengklasifikasikan *trabecular bone*. Hasil dari proses ekstraksi fitur digunakan sebagai atribut data *training*. Sedangkan *groundtruth* yang didapatkan dari proses pemilihan manual oleh ahli digunakan sebagai label kelas.

4. *Postprocessing*

Citra biner hasil klasifikasi diperbaiki menggunakan beberapa proses morfologi yang diperlukan hingga terbentuk citra segmentasi *trabecular bone* dengan performa yang lebih baik.

3.1.4 Analisis Aktor

Aktor yang terdapat pada aplikasi penentuan segmentasi *trabecular bone* ini adalah peneliti di bidang biomedis. Pengguna dapat memasukkan citra dari media penyimpanan di komputer.

3.1.5 Analisis Permasalahan

Untuk mengoptimalkan informasi yang terdapat pada tulang *mandible* (rahang bawah), diperlukan suatu metode segmentasi *trabecular bone* yang akurat. Dengan adanya aplikasi segmentasi

trabecular bone, diharapkan dapat bermanfaat untuk penelitian selanjutnya.

Hasil segmentasi area *trabecular bone* dapat dijadikan dasar dalam perhitungan informasi struktur *trabecular bone*. Informasi struktur *trabecular bone* yang dapat dihitung diantaranya percabangan, luas rongga antartulang dan sebagainya yang berguna dalam identifikasi penyakit, seperti osteoporosis. Hasil segmentasi yang akurat akan memberikan hasil perhitungan informasi struktur *trabecular bone* yang lebih akurat pula.

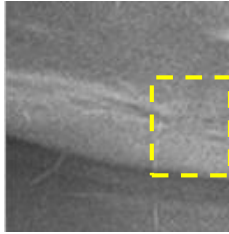
Untuk mencapai hasil segmentasi yang akurat, terdapat beberapa permasalahan yang dapat menurunkan akurasi segmentasi. Permasalahan-permasalahan tersebut dijelaskan sebagai berikut:

ROI yang dipilih pada *dental panoramic radiograph* ini terdapat perbedaan kontras yang pada ROI bagian atas, yang berisikan area *trabecular bone* dan akar gigi, dan ROI bagian bawah, yang berisikan area *trabecular bone*, *cortical bone* dan area di bawah rahang. Perbedaan kontras terlihat di antara area *cortical bone* dan *trabecular bone* seperti ditunjukkan pada Gambar 3.1. Hal ini mengakibatkan segmentasi yang dilakukan secara langsung pada ROI bagian atas dan ROI bagian bawah memberikan hasil yang kurang akurat. Oleh karena itu, diperlukan pemisahan ROI menjadi ROI atas dan ROI bawah.

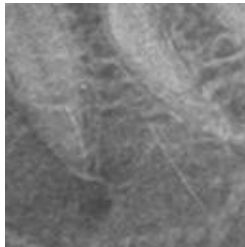
Detail struktur *trabecular bone* yang melintang di area *trabecular bone* seperti ditunjukkan pada Gambar 3.2 dapat mengakibatkan terganggunya proses segmentasi karena intensitas piksel pada struktur *trabecular bone* memiliki warna yang mirip dengan struktur gigi atau *cortical bone*. Jika tidak diperhalus atau dihilangkan, hal ini mengakibatkan proses segmentasi ini menganggap struktur *trabecular bone* tersebut sebagai *cortical bone* atau gigi. Oleh karena itu perlu dilakukan penghilangan struktur *trabecular bone* tersebut.

Pada ROI atas, dilakukan filter mean dan median setelah normalisasi. Hal ini dilakukan karena *noise* dan struktur *trabecular bone* yang ada tidak beraturan. Namun pada ROI bawah,

digunakan filter *motion blur*. Dengan adanya filter *motion blur* ini, *noise* dan struktur *trabecular bone* tersebut akan hilang namun tetap memperkuat batas tiap segmen terutama *cortical bone* yang memisahkan *trabecular bone*.



Gambar 3.1 ROI bawah. Area dalam kotak kuning memperlihatkan rendahnya kontras



Gambar 3.2 Detail Struktur *Trabecular Bone* terdapat diantara dua akar gigi

3.2 Tahap Perancangan

Tahap analisis mendefinisikan kebutuhan yang akan dipenuhi dalam pembangunan aplikasi segmentasi *trabecular bone*.

3.2.1 Perancangan Sistem

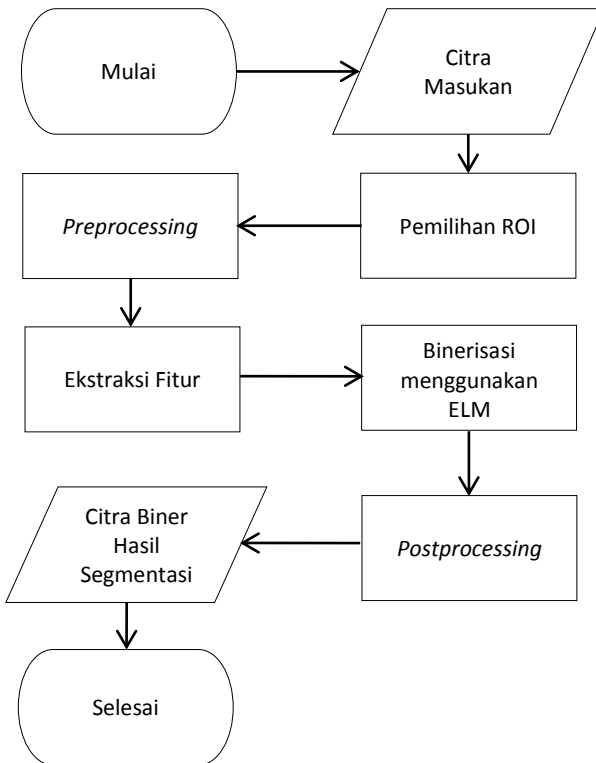
Perancangan sistem dilakukan untuk menggambarkan proses secara keseluruhan dari aplikasi segmentasi *trabecular*

bone. Perancangan sistem dilakukan untuk menggambarkan proses secara keseluruhan dari aplikasi segmentasi *trabecular bone*. Pada aplikasi tersebut terdapat empat proses utama. Proses pertama adalah *preprocessing*. Proses kedua adalah ekstraksi fitur. Proses ketiga adalah segmentasi segmen *trabecular bone*. Proses keempat adalah *postprocessing*. Diagram alir dari sistem ditunjukkan pada Gambar 3.3.

Proses pertama adalah pemilihan ROI, dilakukan secara manual dengan memilih empat area persegi berbeda yaitu dua kanan dan dua kiri dengan ukuran 128x128 piksel pada tulang rahang bawah. Pemilihan ROI tersebut dilakukan pada area *trabecular bone* yang mengandung area gigi untuk ROI pertama dan *cortical bone* pada ROI kedua. Proses berikutnya adalah melakukan *preprocessing*. *Preprocessing* yang dilakukan antara lain normalisasi citra, *median filter*, dan *mean filter*.

Setelah di *preprocessing*, citra tersebut diekstraksi menjadi beberapa fitur pada setiap pikselnya. Ekstraksi fitur merupakan proses yang dilakukan untuk mendapatkan vektor fitur pada tiap piksel citra hasil filter. Ekstraksi fitur pada segmen *trabecular* atas antara lain: intensitas citra, hasil 2-D *gaussian filter*, dan hasil *radial Log Gabor filter*. Sedangkan ekstraksi fitur pada segmen *trabecular* bawah antara lain: intensitas citra, hasil 2-D *Gaussian filter*, hasil *region filling* pada *phase congruency*, dan hasil *region filling* pada *Laplacian of Gaussian*. Nilai dari vektor fitur tersebut digunakan dalam proses pembangunan metode klasifikasi *Extreme Learning Machine* menggunakan *training data*.

Proses binerisasi *trabecular bone* merupakan proses yang dilakukan untuk mendapatkan citra hitam putih dari ROI *trabecular bone*. Proses segmentasi didahului dengan pemilihan *training data* menggunakan metode klasifikasi *Extreme Learning Machine*. Selanjutnya dilanjutkan dengan proses *testing*, yaitu pengelompokkan piksel menggunakan metode klasifikasi tersebut dengan memanfaatkan vektor fitur dari proses ekstraksi fitur. Citra biner hasil klasifikasi kemudian diperbaiki segmennya pada tahap *postprocessing* sehingga terbentuk citra akhir hasil segmentasi.



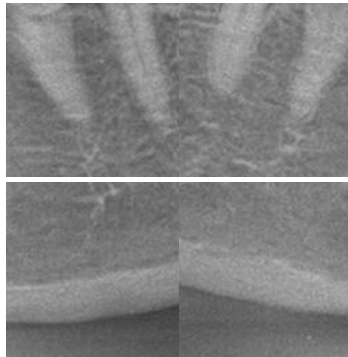
Gambar 3.3 Diagram alir dari sistem aplikasi segmentasi *trabecular bone*.

3.2.2 Perancangan Data

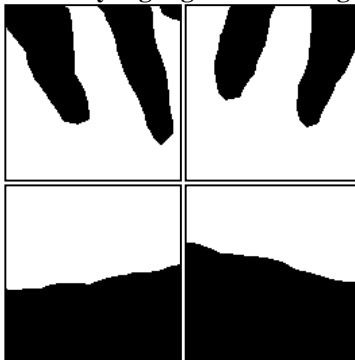
Perancangan data dilakukan untuk memastikan pengoperasian aplikasi berjalan dengan benar. Data yang diperlukan dalam pengoperasian aplikasi adalah data masukan (*input*) dan data keluaran (*output*) yang akan digunakan oleh pengguna.

3.2.2.1 Data Masukan

Data masukan merupakan data yang diolah menggunakan aplikasi segmentasi *trabecular bone*. Data yang digunakan sebagai masukan adalah citra *dental panoramic radiograph* dari pasien perempuan dengan ras Asia (Indonesia). Dari citra *dental panoramic radiograph* dipilih ROI dilakukan secara manual dengan memilih empat area persegi berbeda yaitu dua kanan dan dua kiri dengan ukuran 128x128 piksel pada tulang rahang bawah pada area *trabecular bone* yang mengandung area gigi untuk ROI pertama dan *cortical bone* pada ROI kedua. Contoh citra yang digunakan sebagai data masukan ditunjukkan pada Gambar 3.4.



Gambar 3.4 ROI yang digunakan sebagai input



Gambar 3.5 citra output yang diharapkan.

3.2.2.2 Data Keluaran

Data keluaran dari aplikasi segmentasi *trabecular bone* merupakan citra hitam putih dengan ROI *trabecular bone*. Warna putih menunjukkan *trabecular bone* dan warna hitam menunjukkan *non-trabecular bone*. Citra tersebut ditunjukkan pada Gambar 3.5.

3.2.3 Perancangan Proses

Perancangan proses dilakukan untuk memberikan gambaran mengenai setiap proses yang terdapat pada aplikasi segmentasi *trabecular bone*

3.2.3.1 Preprocessing

Dalam mengawali proses pengerjaan program, maka harus disiapkan terlebih dahulu data yang akan digunakan dalam proses pengerjaan. *Preprocessing* adalah tahapan yang penting untuk dilakukan. Pada tugas akhir kali ini, tahap *preprocessing* yang dilakukan adalah pemilihan ROI, normalisasi citra, filter mean, filter median, dan filter *motion blur*. Diagram alir dari proses ditunjukkan pada Gambar 3.6. Pemilihan ROI dalam penelitian ini dilakukan dua kali pada tiap citra dengan ukuran 256 x 128, yaitu di bagian kiri dan kanan. ROI yang terpilih masing-masing mengandung gigi dan *cortical bone*. Kemudian ROI tersebut dibagi lagi menjadi dua dengan ukuran 128 x 128, yaitu bagian atas dan bawah. Normalisasi citra dilakukan dengan tujuan untuk menyamakan rentang keabuan antara satu citra dengan citra lainnya. Filter mean dan filter median digunakan untuk menghilangkan noise pada citra gigi. Pada ROI atas, dilakukan filter mean dan median setelah normalisasi. Hal ini dilakukan karena *noise* yang ada tidak beraturan. Namun pada ROI bawah, digunakan filter *motion blur*. Hal ini dikarenakan *noise* pada ROI ini cenderung searah dengan *cortical bone*. Dengan adanya filter *motion blur* ini, *noise* tersebut akan hilang namun tetap memperkuat batas tiap segmen terutama *cortical bone* yang memisahkan *trabecular bone* dengan latar belakang. Hasil *preprocessing* dipaparkan pada Gambar 3.7.

3.2.3.2 Ekstraksi Fitur

Proses ekstraksi fitur dilakukan untuk mendapatkan vektor fitur yang digunakan dalam proses klasifikasi segmentasi. Dalam tugas akhir ini ada empat metode yang digunakan, yaitu: Intesitas piksel citra, filter *Gaussian* 2D, filter *Log-Gabor*, dan *phase congruency*.

3.2.3.2.1 Intesitas piksel citra

Pada citra ROI yang telah di *preprocess*, segmen *trabecular bone* lebih gelap dari pada citra gigi untuk ROI *trabecular* atas. Pada ROI bawah, segmen *trabecular bone* lebih terang daripada latar belakang namun lebih gelap daripada *cortical bone*. Oleh karena itu, piksel citra ROI ini digunakan sebagai fitur pertama. Diagram alir proses ini dipaparkan pada Gambar 3.8.

3.2.3.2.2 Filter Gaussian 2D

Ekstraksi fitur selanjutnya adalah citra hasil filter *Gaussian* 2D. Filter ini digunakan untuk memperjelas wilayah tiap segmen. Langkah pertama adalah membaca citra yang telah di *preprocessing*. Langkah kedua adalah membangun *kernel Gaussian* 2D. Langkah ketiga adalah meng-konvolusi menggunakan *kernel Gaussian* 2D. Parameter sigma yang dipakai pada tugas akhir ini adalah $\sqrt{2}$ dan 2. Diagram alir dan hasil output dipaparkan masing-masing pada Gambar 3.9 dan Gambar 3.10.

3.2.3.2.3 Filter Log Gabor

Dalam proses ekstraksi fitur filter *Log Gabor* ini untuk memperjelas segmen pada Transformasi Fourier. Ekstraksi fitur ini hanya digunakan untuk ROI atas. Filter *Log Gabor* yang dipakai pada metode ini adalah *Radial Log Gabor*. Langkah pertama adalah membaca citra yang telah di *pre processing*. Langkah kedua membangun *kernel radial Log Gabor* dan men-transformasi Fourier citra ROI. Langkah ketiga adalah mendapatkan citra yang telah dibangun dengan filter *Radial Log Gabor*. Parameter yang digunakan adalah *wavelength* sebesar 10 dan sigma sama dengan

10. Diagram alir dan hasil output masing-masing dipaparkan di Gambar 3.11 dan Gambar 3.12.

3.2.3.2.4 Phase Congruency

Ekstraksi fitur ini digunakan untuk memberikan batas antar segmen. Ekstraksi fitur ini hanya digunakan untuk ROI bawah. Kemudian kita pilih segmen *trabecular bone* yaitu segmen paling atas. Langkah pertama adalah membaca citra ROI bawah yang telah *preprocessing*. Kemudian pada langkah kedua adalah merancang filter *Angular Log Gabor* pada tiap orientasi dan mentransformasi citra ROI ke bentuk Fourier. Langkah ketiga adalah mendapatkan citra yang telah dibangun menggunakan filter *Angular Log Gabor*. Lalu langkah keempat adalah mendapatkan nilai penjumlahan amplitudo, *even filter*, dan *odd filter* pada tiap orientasi. Langkah kelima adalah menghitung nilai *phase congruency* tiap piksel.

Hasil *phase congruency* ini kemudian di-binerisasi dengan parameter 0.3. Langkah ketujuh adalah mendapatkan *edge cortical bone* paling atas. Langkah selanjutnya adalah mengisi segmen diatas *edge* tersebut. Diagram alir dan hasil output dipaparkan masing-masing pada Gambar 3.13 dan Gambar 3.14.

3.2.3.2.5 Laplacian of Gaussian

Ekstraksi fitur ini digunakan untuk memberikan batas antar segmen. Ekstraksi fitur ini hanya digunakan untuk ROI bawah. Kemudian kita pilih segmen *trabecular bone* yaitu segmen paling atas. Langkah pertama adalah membaca citra yang telah di *pre processing*. Langkah kedua mendapatkan citra hasil *Laplacian of Gaussian*. Langkah ketiga adalah mendapatkan *edge cortical bone* paling atas. Langkah keempat adalah mengisi segmen diatas *edge* tersebut. Diagram alir dan hasil output dipaparkan masing-masing pada Gambar 3.15 dan Gambar 3.16.

3.2.3.3 Klasifikasi segmen trabecular bone

Setelah mendapatkan ekstraksi fitur, dilanjutkan ke proses klasifikasi menggunakan *Extreme Learning Machine*.

Diambil sampel sebanyak 22184 data yang diambil dari delapan citra ROI. Sampel ini digunakan sebagai data latih pada metode klasifikasi ini.

Model ELM ini selanjutnya digunakan pada proses *testing* (klasifikasi piksel) di mana satu kali proses binerisasi mengelompokkan 16.384 data. Pengelompokkan piksel dengan ELM ini menghasilkan dua kelas, yaitu piksel putih (positif) dan piksel hitam (negatif). Diagram dan hasil binerisasi citra dari masing-masing ditunjukkan pada Gambar 3.17 dan Gambar 3.18.

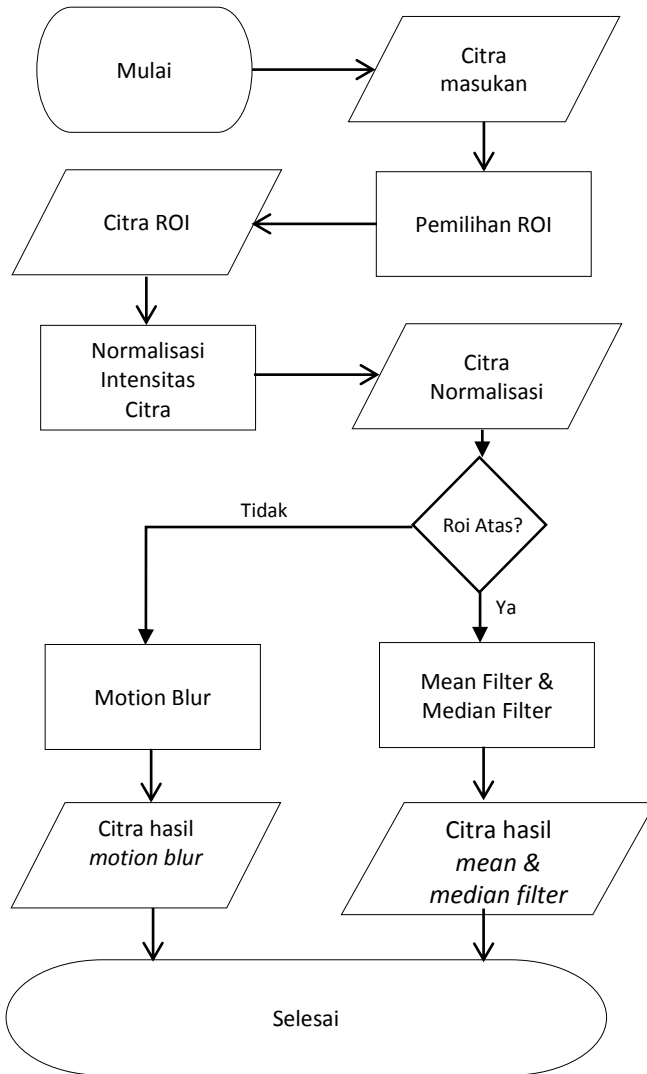
3.2.3.4 Postprocessing

Postprocessing adalah perbaikan segmen pada citra biner. Operasi yang pertama kali dilakukan adalah memperhalus hasil segmentasi menggunakan filter median. Kemudian dilanjutkan dengan pengambilan area besar menggunakan fitur matlab *regionprops*. Operasi ini dilakukan untuk membersihkan *noise* dan mengisi *hole* dimana *noise* dan *hole* tersebut memiliki area yang bernilai kecil. Diagram alir dan hasil perbaikan segmen citra biner masing-masing ditunjukkan pada Gambar 3.19 dan Gambar 3.20.

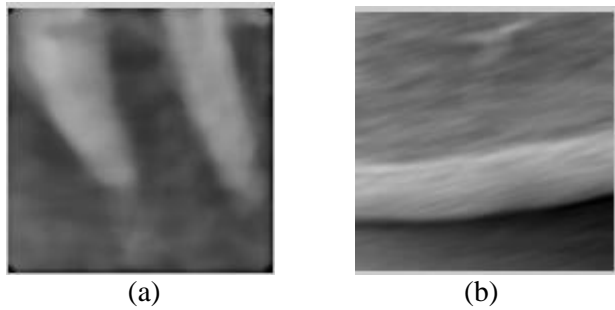
3.2.3.5 Rancangan Antarmuka

Rancangan antarmuka pada tugas akhir ini berupa citra *dental panoramic radiograph* sebagai masukan, empat citra ROI dan empat citra hasil segmentasi. Visualisasi tampilan antarmuka aplikasi ditunjukkan pada Gambar 3.21.

Pada kotak paling besar, terdapat citra *dental panoramic radiograph*. Di bagian bawah kotak tersebut, terdapat tombol “Pilih Berkas” untuk memuat citra *dental panoramic radiograph*. Kemudian terdapat tombol “Pilih ROI” untuk menampilkan kotak ROI berukuran 256 x 128. Di samping kanan citra terdapat masing-masing empat pasang kotak. Mulai dari atas adalah citra ROI dan citra hasil segmentasi.

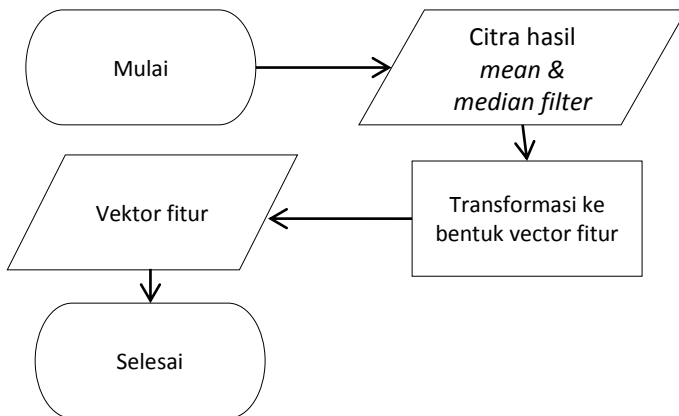


Gambar 3.6 Diagram Alir Tahap *Preprocessing*

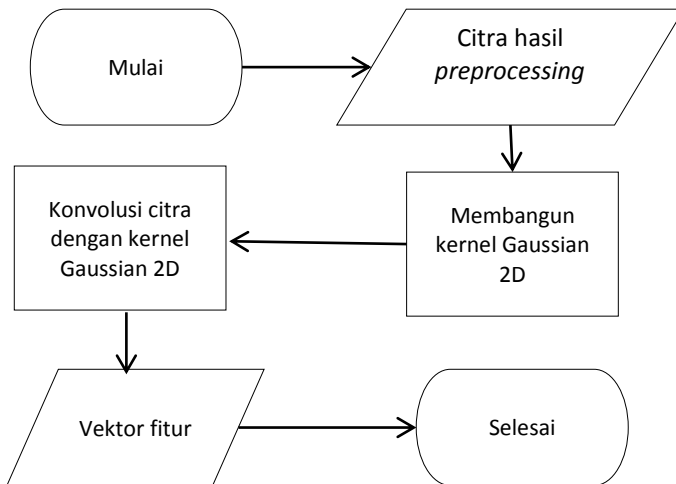


Gambar 3.7 Hasil *preprocessing*

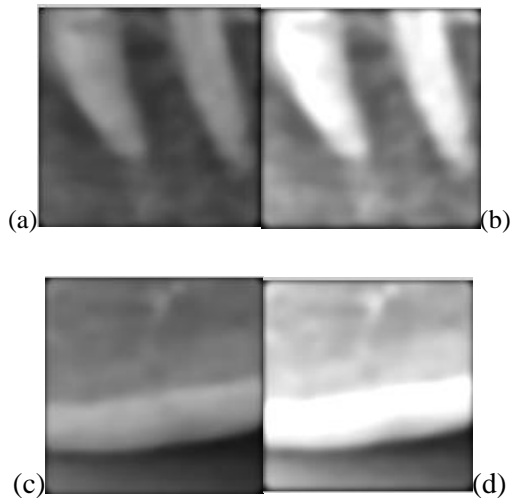
(a) hasil mean & median filter, (b) hasil *motion blur*



Gambar 3.8 Diagram Alir Pengambilan fitur intensitas citra

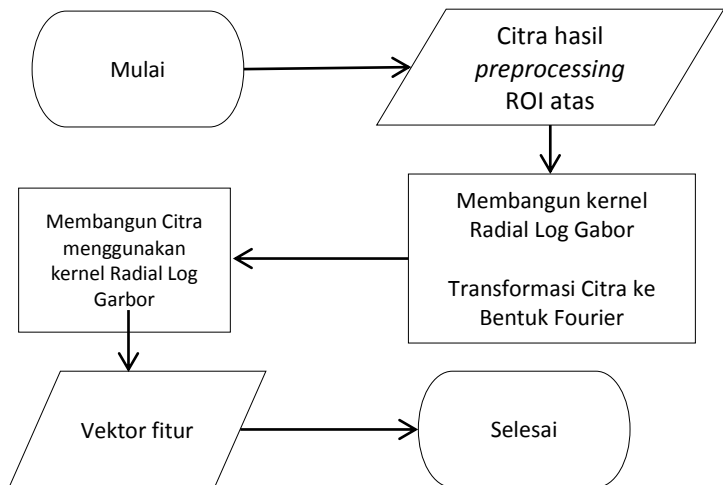


Gambar 3.9 Diagram Alir Filter Gaussian 2D



Gambar 3.10 Citra Hasil Ekstraksi Fitur Gaussian 2D

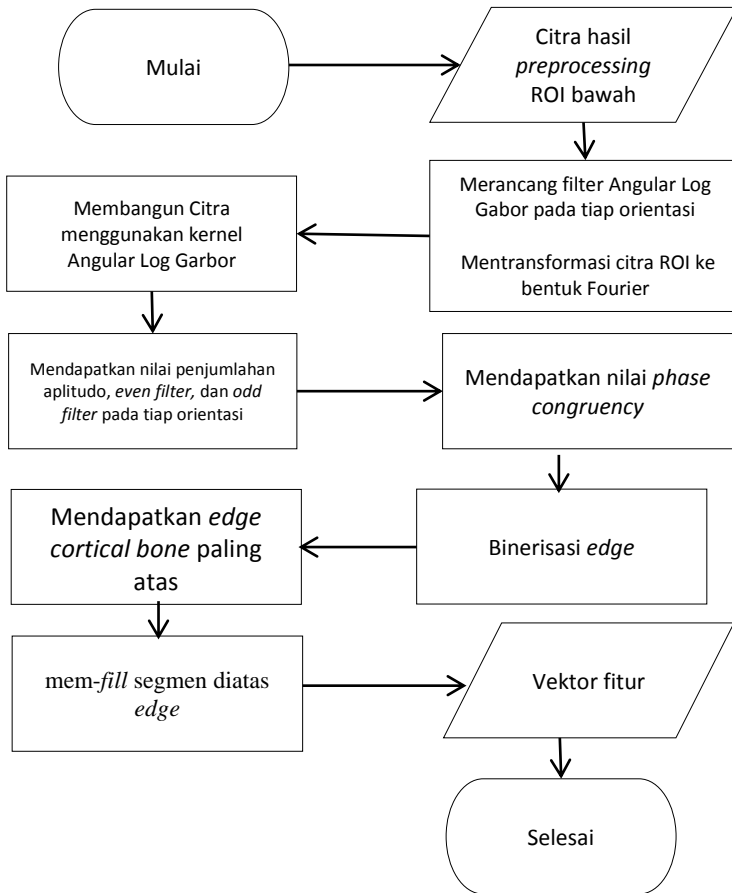
(a) & (c) menggunakan $\sigma = \sqrt{2}$, (b) & (d) menggunakan $\sigma = 2$



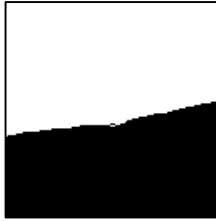
Gambar 3.11 Diagram Alir Ekstraksi Fitur Filter Log Gabor



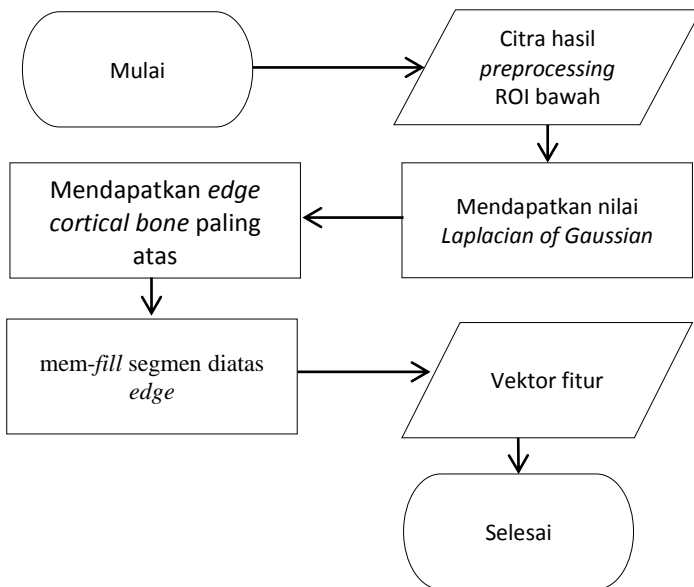
Gambar 3.12 Hasil Akhir Ekstraksi Fitur Filter Log Gabor



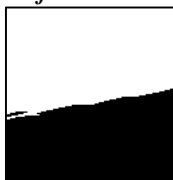
Gambar 3.13 Diagram alir proses ekstraksi fitur dengan *phase congruency*



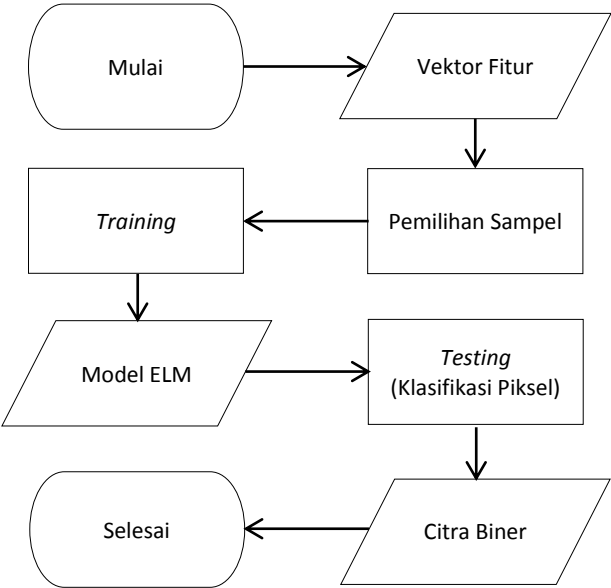
Gambar 3.14 Hasil Output *Phase Congruency*



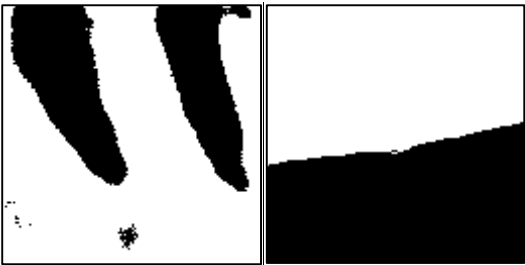
Gambar 3.15 Diagram alir proses ekstraksi fitur dengan *Laplacian of Gaussian*



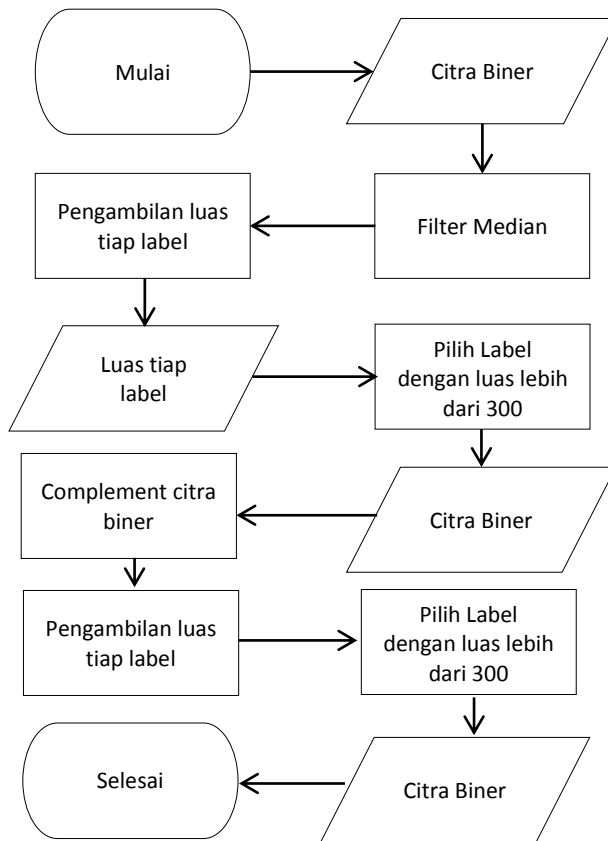
Gambar 3.16 Hasil Output *Laplacian of Gaussian*



Gambar 3.17 Diagram Alir Klasifikasi segmen menggunakan ELM



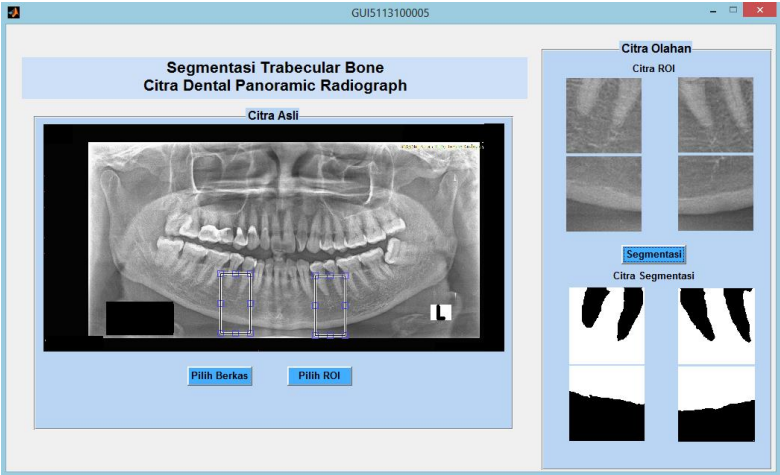
Gambar 3.18 Citra hasil segmentasi



Gambar 3.19 Diagram alir *postprocessing*



Gambar 3.20 Citra hasil *postprocessing*



Gambar 3.21 Rancangan Antarmuka

BAB IV IMPLEMENTASI

Pada bab ini diuraikan mengenai implementasi perangkat lunak dari rancangan metode yang telah dibahas pada Bab III yang meliputi kode program dalam perangkat lunak. Selain itu, dijelaskan pula mengenai implementasi dari tiap proses, parameter masukan, keluaran, dan beberapa keterangan yang berhubungan dengan program.

4.1 Lingkungan Implementasi

Dalam implementasi algoritma digunakan perangkat-perangkat sebagai berikut:

4.1.1 Perangkat Keras

Berikut spesifikasi perangkat keras yang digunakan saat implementasi:

- Perangkat : Lenovo H530S
- Prosesor : Intel® Core™ i3 4150 CPU @ 3,5GHz
- Memori : 4.00 GB

4.1.2 Perangkat Lunak

Berikut perangkat lunak yang digunakan dalam implementasi:

1. Windows 8.1
2. Matlab R2014a
3. GIMP 2.8.18
4. Microsoft Excel

4.2 Implementasi Tahap *Preprocessing*

Tahap *preprocessing* pada tugas akhir ini terdiri dari dua langkah. Langkah pertama adalah normalisasi. Pada langkah kedua dilakukan *preprocessing* untuk ROI atas dan bawah yang berbeda.

4.2.1 Normalisasi Intensitas Citra

Normalisasi intensitas citra dilakukan dengan tujuan untuk menyamakan rentang keabuan antara satu citra dengan citra lainnya. Formulasi normalisasi ditunjukkan pada Persamaan(2.1). Implementasi ini mengikuti diagram alir yang telah dipaparkan pada Gambar 3.6. Implementasi fungsi normalisasi ditunjukkan pada Kode Sumber 4.1

1	<code>function imNew=normalisasi(I)</code>
2	<code>imNew=(I-min(I(:))) * double(255/(max(I(:)) - min(I(:))));</code>
3	<code>end</code>

Kode Sumber 4.1 Implementasi Fungsi Normalisasi Citra

4.2.2 Implementasi *Preprocessing* ROI atas

Implementasi ini mengikuti diagram alir yang telah dipaparkan pada Gambar 3.6. *Preprocessing* ini dilakukan normalisasi terlebih dahulu. Kemudian dilakukan filter median dengan matrix 10x10. Setelah itu dilakukan filter mean dengan parameter *default* dari Matlab yaitu 3x3. Kemudian dilakukan normalisasi lagi. Implementasi *preprocessing* ROI bawah ini ditunjukkan pada Kode Sumber 4.2.

1	<code>imgori=normalisasi(imgori);</code>
2	<code>imgori=medfilt2(imgori, [10 10]);</code>
3	<code>H2=fspecial('average');</code>
4	<code>imgori=imfilter(imgori,H2,'replicate');</code>
5	<code>imNew=normalisasi(imgori);</code>

Kode Sumber 4.2 Implementasi fungsi *preprocessing* ROI

atas

4.2.3 Implementasi *Preprocessing* ROI bawah

Implementasi ini mengikuti diagram alir yang telah dipaparkan pada Gambar 3.6. *Preprocessing* ini dilakukan normalisasi terlebih dahulu. Kemudian dilakukan *motion* filter dengan matriks arah -15 derajat untuk ROI kiri dan 15 derajat

untuk ROI kanan. Terakhir dilakukan normalisasi. Implementasi *preprocessing* ROI bawah ini ditunjukkan pada Kode Sumber 4.3.

1	<code>imgori=normalisasi(imgori);</code>
2	<code>if mod(i,2)==0</code>
3	<code> H3=fspecial('motion',15,-15);</code>
4	<code>else</code>
5	<code> H3=fspecial('motion',15,15);</code>
6	<code>imgori=imfilter(imgori,H3,'replicate');</code>
7	<code>imNew=normalisasi(imgori);</code>

Kode Sumber 4.3 Implementasi fungsi *preprocessing* ROI bawah

4.3 Implementasi Proses Ekstraksi Fitur

Pada implementasi proses ekstraksi fitur terdapat empat metode yang digunakan. Metode-metode tersebut yaitu filter *Gaussian* 2D, filter *Log Gabor*, dan *Phase Congruency*.

4.3.1 Implementasi Proses Filter Gaussian 2D

Implementasi filter *Gaussian* 2D ini adalah membaca citra yang telah di *preprocessing* yang dipaparkan pada baris 1. Langkah kedua adalah membangun *kernel Gaussian* 2D yang dipaparkan pada baris 2 sampai 5. Langkah ketiga adalah meng-konvolusi menggunakan *kernel Gaussian* 2D yang dipaparkan pada baris 6. Parameter sigma yang dipakai pada tugas akhir ini adalah $\sqrt{2}$ dan 2. Implementasi ini mengikuti diagram alir yang telah dipaparkan pada Gambar 3.9. Implementasi *preprocessing* ROI bawah ini ditunjukkan pada Kode Sumber 4.4.

1	<code>I = double(image);</code>
2	<code>sz = 3;</code>
3	<code>[x,y]=meshgrid(-sz:sz,-sz:sz);</code>
4	<code>Exp_comp = -(x.^2+y.^2)/(2*sigma*sigma);</code>
5	<code>Kernel= exp(Exp_comp)/(2*pi*sigma*sigma);</code>
6	<code>Output=imfilter(I,Kernel,'replicate');</code>

Kode Sumber 4.4 Implementasi filter Gaussian 2D.

4.3.2 Implementasi Proses Filter Log Gabor

Implementasi proses Filter *Log Gabor* ini hanya digunakan untuk ROI atas. Implementasi ini mengikuti diagram alir yang telah dipaparkan pada Gambar 3.11. Langkah pertama adalah membaca citra yang telah di *preprocessing*. Langkah kedua membangun kernel radial *Log Gabor* yang dipaparkan pada baris 11 sampai 13 dan men-transformasi *Fourier* citra ROI yang dipaparkan pada baris 1 sampai 9. Langkah ketiga adalah mendapatkan citra yang telah dibangun dengan filter Radial *Log Gabor* yang dipaparkan pada baris 15. Parameter yang digunakan adalah *wavelength* sebesar 10 dan sigma sama dengan 10. Implementasi proses filter *Log Gabor* ditunjukkan pada Kode Sumber 4.5.

1	ep=0.00001;
2	[rows, cols] = size(I);
3	[ux, uy] = meshgrid(([1:cols] - (fix(cols/2)+1)) / (cols-mod(cols,2)), ...
4	([1:rows] - (fix(rows/2)+1)) / (rows-mod(rows,2)));
5	ux = ifftshift(ux); % Quadrant shift to put 0 frequency at the corners
6	uy = ifftshift(uy);
7	% Convert to polar coordinates
8	th = atan2(uy,ux);
9	r = sqrt(ux.^2 + uy.^2);
10	% Create spectrum
11	filterFFT = 1.0/wavelength;
12	filterFFT = exp((- (log(double(r/filterFFT + ep))) .^2) / (2 * log(sigma)^2));
13	filterFFT(1,1) = 0;
14	% Filter image
15	I_filtered = real(ifft2(fft2(I) .* filterFFT)) + imag(ifft2(fft2(I) .* filterFFT));

Kode Sumber 4.5 Implementasi fungsi proses filter Log Gabor.

4.3.3 Implementasi Proses *Phase Congruency*

4.3.3.1 Implementasi Proses Membuat *Edge* menggunakan *Phase Congruency*

Implementasi proses ini mengikuti diagram alir yang telah dipaparkan pada Gambar 3.13. Implementasi proses *phase congruency* dimulai dengan membaca citra ROI bawah yang telah *preprocessing*. Kemudian pada langkah kedua adalah merancang filter *Angular Log Gabor* pada tiap orientasi yang dipaparkan pada baris 11 sampai 46 dan mentransformasi citra ROI ke bentuk Fourier yang dipaparkan pada baris 1. Langkah ketiga adalah mendapatkan citra yang telah dibangun menggunakan filter *Angular Log Gabor* yang dipaparkan pada baris 47. Lalu langkah keempat adalah mendapatkan nilai penjumlahan amplitudo, *even filter*, dan *odd filter* pada tiap orientasi yang dipaparkan pada baris 48 sampai 80. Langkah kelima adalah menghitung nilai *phase congruency* tiap piksel yang dipaparkan pada baris 84 sampai 139. Hasil dari Kode sumber proses ini ditunjukkan pada Kode Sumber 4.6.

1	<code>imagefft = fft2(im);</code>
2	<code>sze = size(imagefft);</code>
3	<code>rows = sze(1);</code>
4	<code>cols = sze(2);</code>
5	<code>zero = zeros(sze);</code>
6	<code>totalEnergy = zero;</code>
7	<code>totalSumAn = zero;</code>
8	<code>orientation = zero;</code>
9	<code>estMeanE2n = [];</code>
10	
11	<code>x = ones(rows,1) * (-cols/2 : (cols/2 - 1))/(cols/2);</code>
12	<code>y = (-rows/2 : (rows/2 - 1))' * ones(1,cols)/(rows/2);</code>
13	<code>radius = sqrt(x.^2 + y.^2</code>
14	<code>radius(round(rows/2+1),round(cols/2+1)) = 1;</code>
15	<code>theta = atan2(-y,x);</code>
16	<code>sintheta = sin(theta);</code>
17	<code>costheta = cos(theta);</code>
18	<code>clear x; clear y; clear theta;</code>

19	<code>for o = 1:norient</code>
20	<code> angl = (o-1)*pi/norient;</code>
21	<code> wavelength = minWaveLength;</code>
22	<code> sumE ThisOrient = zero;</code>
23	<code> sumO ThisOrient = zero;</code>
24	<code> sumAn ThisOrient = zero;</code>
25	<code> Energy ThisOrient = zero;</code>
26	<code> EOArray = [];</code>
27	<code> ifftFilterArray =</code>
28	
29	<code> ds = sintheta * cos(angl) - costheta * sin(angl);</code>
30	<code> dc = costheta * cos(angl) + sintheta * sin(angl);</code>
31	<code> dtheta = abs(atan2(ds,dc));</code>
32	<code> spread = exp((-dtheta.^2) / (2 * thetaSigma^2));</code>
33	
34	<code> for s = 1:nscale,</code>
35	<code> fo = 1.0/wavelength;</code>
36	<code> rfo = fo/0.5;</code>
37	
38	<code> logGabor = exp((-log(radius/rfo)).^2) / (2 * log(sigmaOnf)^2));</code>
39	<code>logGabor(round(rows/2+1),round(cols/2+1)) = 0;</code>
40	
41	<code> filter = logGabor .* spread;</code>
42	<code> filter = fftshift(filter);</code>
43	
44	<code> ifftFilt = real(ifft2(filter)) * sqrt(rows*cols);</code>
45	<code> ifftFilterArray = [ifftFilterArray ifftFilt];</code>
46	
47	<code> EOfft = imagefft .* filter;</code>
48	<code> EO = ifft2(EOfft);</code>
49	
50	<code> EOArray = [EOArray, EO];</code>
51	<code> An = abs(EO);</code>
52	

53	<code>sumAn_ThisOrient = sumAn_ThisOrient +</code> <code>An;</code>
54	<code>sumE_ThisOrient = sumE_ThisOrient +</code> <code>real(EO);</code>
55	<code>sumO_ThisOrient = sumO_ThisOrient +</code> <code>imag(EO);</code>
56	
57	<code>if s == 1</code>
58	<code>maxAn = An;</code>
59	<code>else</code>
60	<code>maxAn = max(maxAn, An);</code>
61	<code>end</code>
62	
63	<code>if s==1</code>
64	<code>EM n = sum(sum(filter.^2));</code>
65	<code>end</code>
66	
67	<code>wavelength = wavelength * mult;</code>
68	<code>end</code>
69	
70	
71	<code>XEnergy = sqrt(sumE_ThisOrient.^2 +</code> <code>sumO_ThisOrient.^2) + epsilon;</code>
72	<code>MeanE = sumE_ThisOrient ./ XEnergy;</code>
73	<code>MeanO = sumO_ThisOrient ./ XEnergy;</code>
74	
75	
76	<code>for s = 1:nscale,</code>
77	<code>EO = submat(EOArray,s,cols);</code>
78	<code>E = real(EO); O = imag(EO);</code>
79	<code>Energy_ThisOrient = Energy_ThisOrient</code> <code>...</code>
80	<code>+ E.*MeanE + O.*MeanO - abs(E.*MeanO</code> <code>- O.*MeanE);</code>
81	<code>end</code>
82	
83	
84	<code>medianE2n =</code> <code>median(reshape(abs(submat(EOArray,1,cols)).^2,1,ro</code> <code>ws*cols));</code>
85	<code>meanE2n = -medianE2n/log(0.5);</code>
86	<code>estMeanE2n = [estMeanE2n meanE2n];</code>

87	
88	noisePower = meanE2n/EM_n;
89	
90	EstSumAn2 = zero;
91	for s = 1:nscale
92	EstSumAn2 =
	EstSumAn2+submat(ifftFilterArray,s,cols).^2;
93	end
94	
95	EstSumAiAj = zero;
96	for si = 1:(nscale-1)
97	for sj = (si+1):nscale
	EstSumAiAj = EstSumAiAj +
98	submat(ifftFilterArray,si,cols).*submat(ifftFilter
	Array,sj,cols);
99	end
100	end
101	
102	EstNoiseEnergy2 = 2*noisePower*sum(sum(EstSumAn2))
	+ 4*noisePower*sum(sum(EstSumAiAj));
103	
104	tau = sqrt(EstNoiseEnergy2/2);
105	EstNoiseEnergy = tau*sqrt(pi/2);
106	EstNoiseEnergySigma = sqrt((2-pi/2)*tau^2
);
107	
108	T = EstNoiseEnergy +
	k*EstNoiseEnergySigma;
109	
110	T = T/1.7;
111	
112	Energy_ThisOrient = max(Energy_ThisOrient
	- T, zero);
113	
114	width = sumAn_ThisOrient ./ (maxAn +
	epsilon) / nscale;
115	
116	weight = 1.0 ./ (1 + exp((cutOff -
	width)*g));
117	
118	Energy_ThisOrient =
	weight.*Energy_ThisOrient;

119	
120	totalSumAn = totalSumAn + sumAn ThisOrient;
121	totalEnergy = totalEnergy + Energy ThisOrient;
122	
123	if(o == 1),
124	maxEnergy = Energy ThisOrient;
125	featType = E + i*O;
126	else
127	change = Energy ThisOrient > maxEnergy;
128	orientation = (o - 1).*change + orientation.*(~change);
129	featType = (E+i*O).*change + featType.*(~change);
130	maxEnergy = max(maxEnergy, Energy ThisOrient);
131	end
132	
133	end
134	
135	phaseCongruency = totalEnergy ./ (totalSumAn + epsilon);
136	
137	orientation = orientation * (180 / norient);
138	
139	featType = featType*i;

Kode Sumber 4.6 Implementasi *phase congruency*.

4.3.3.2 Implementasi Proses Membuat *Region* hasil *Phase Congruency*

Implementasi proses ini mengikuti diagram alir yang telah dipaparkan pada Gambar 3.13. Hasil *phase congruency* ini kemudian di-binerisasi dengan parameter pembatas 0.3 yang dipaparkan pada baris 3. Langkah ketujuh adalah mendapatkan *edge cortical bone* paling atas yang dipaparkan pada baris 4 sampai 36. Langkah selanjutnya adalah mengisi segmen diatas *edge* tersebut yang dipaparkan pada baris 44 sampai 59. Hasil dari Kode

sumber proses *region* hasil *phase congruency* ditunjukkan pada Kode Sumber 4.7.

1	<code>addpath('Gabor_Image_features/Gabor_Image_featu res');</code>
2	<code>pc=phasecong(image,3,6,5);</code>
3	<code>bw=im2bw(pc,0.3);</code>
4	<code>bw(:,1:4)=0;</code>
5	<code>bw(:,125:129)=0;</code>
6	<code>bw(1:4,⊙)=0;</code>
7	<code>bw(125:129,⊙)=0;</code>
8	<code>bw=bwmorph(bw,'thin',inf);</code>
9	<code>if mod(i,2)==1</code>
10	<code>orient=-15;</code>
11	<code>else</code>
12	<code>orient=15;</code>
13	<code>end</code>
14	<code>s=strel('line',15,orient);</code>
15	<code>bw=imclose(bw,s);</code>
18	<code>bwL=bwlabel(bw);</code>
19	
20	<code>stats=regionprops(bw,'MajorAxisLength','Centroi d','PixelList');</code>
21	<code>[length, idx] = sort([stats.MajorAxisLength],'descend');</code>
22	
23	<code>pxList={stats.PixelList};</code>
24	<code>56enter=zeros(1,2);</code>
25	<code>for iter=1:2</code>
26	<code>pxListIdx=pxList{idx(iter)};</code>
27	<code>idxIter=find(pxListIdx(:,1)==64);</code>
28	<code>56enter(iter)=pxListIdx(idxIter(1),2);</code>
29	<code>end</code>

30	
31	[length, idx2]=sort(57enter);
32	
33	imOutput=ismember(bwL,idx(idx2(1)));
34	
35	indexChooosen=idx(idx2(1));
36	pxListChooosen=pxList{indexChooosen};
37	
38	filllList=zeros(129,1);
39	for iterList=1:size(pxListChooosen,1)
40	imOutput(1:pxListChooosen(iterList,2),pxListChooosen(iterList,1)) = 1;
41	filllList(pxListChooosen(iterList,1))=pxListChooosen(iterList,2);
42	end
43	
44	yToLeft=0;
45	yToRight=0;
46	for iterFill=1:size(filllList,1)
47	if filllList(iterFill)==0
48	filllList(iterFill)=yToLeft;
49	imOutput(1:filllList(iterFill),iterFill)=1;
50	end
51	if filllList(size(filllList,1)-iterFill+1)==0
52	filllList(size(filllList,1)-iterFill+1)=yToRight;
53	imOutput(1:filllList(size(filllList,1)-iterFill+1),size(filllList,1)-iterFill+1)=1;
54	end
55	yToLeft=filllList(iterFill);
56	yToRight=filllList(size(filllList,1)-iterFill+1);
57	end

58	
59	<code>figure,imshow(imOutput);</code>

Kode Sumber 4.7 Implementasi mendapatkan *region* menggunakan *phase congruency*.

4.3.4 Implementasi Membuat *Region* hasil *Laplacian of Gaussian*

Implementasi proses ini mengikuti diagram alir yang telah dipaparkan pada Gambar 3.15. Ekstraksi fitur ini digunakan untuk memberikan batas antar segmen. Ekstraksi fitur ini hanya digunakan untuk ROI bawah. Area yang dipilih adalah segmen *trabecular bone* yaitu segmen paling atas. Langkah pertama adalah membaca citra yang telah di *pre processing*. Langkah kedua mendapatkan citra hasil *Laplacian of Gaussian* yang dipaparkan pada baris 2. Langkah ketiga adalah mendapatkan *edge cortical bone* paling atas yang dipaparkan pada baris 8 sampai 42. Langkah keempat adalah mengisi segmen diatas *edge* tersebut yang dipaparkan pada baris 44 sampai 57. Hasil dari kode sumber proses *region* hasil *laplacian of Gaussian* ditunjukkan pada Kode Sumber 4.8.

1	<code>addpath('Gabor_Image_features/Gabor_Image_featu res');</code>
2	<code>LoG=edge(image, 'log');</code>
3	<code>bw=LoG;</code>
4	<code>bw(:,1:4)=0;</code>
5	<code>bw(:,125:129)=0;</code>
6	<code>bw(1:4,⊙)=0;</code>
7	<code>bw(125:129,⊙)=0;</code>
8	<code>bw=bwmorph(bw,'thin',inf);</code>
9	<code>if mod(i,2)==1</code>
10	<code>orient=-15;</code>
11	<code>else</code>
12	<code>orient=15;</code>
13	<code>end</code>

14	s=strel('line',15,orient);
15	bw=imclose(bw,s);
18	bwL=bwlabel(bw);
19	
20	stats=regionprops(bw,'MajorAxisLength','Centroid','PixelList');
21	[length, idx] = sort([stats.MajorAxisLength],'descend');
22	
23	pxList={stats.PixelList};
24	59enter=zeros(1,2);
25	for iter=1:2
26	pxListIdx=pxList{idx(iter)};
27	idxIter=find(pxListIdx(:,1)==64);
28	59enter(iter)=pxListIdx(idxIter(1),2);
29	end
30	
31	[length, idx2]=sort(centerY);
32	
33	imOutput=ismember(bwL,idx(idx2(1)));
34	
35	indexChoosen=idx(idx2(1));
36	pxListChoosen=pxList{indexChoosen};
37	
38	fillList=zeros(129,1);
39	for iterList=1:size(pxListChoosen,1)
40	imOutput(1:pxListChoosen(iterList,2),pxListChoosen(iterList,1)) = 1;
41	fillList(pxListChoosen(iterList,1))=pxListChoosen(iterList,2);
42	end
43	
44	yToLeft=0;
45	yToRight=0;

46	<code>for iterFill=1:size(fillList,1)</code>
47	<code>if fillList(iterFill)==0</code>
48	<code>fillList(iterFill)=yToLeft;</code>
49	<code>imOutput(1:fillList(iterFill),iterFill)=1</code> <code>;</code>
50	<code>end</code>
51	<code>if fillList(size(fillList,1)-iterFill+1)==0</code>
52	<code>fillList(size(fillList,1)-</code> <code>iterFill+1)=yToRight;</code>
53	<code>imOutput(1:fillList(size(fillList,1)-</code> <code>iterFill+1),size(fillList,1)-iterFill+1)=1;</code>
54	<code>end</code>
55	<code>yToLeft=fillList(iterFill);</code>
56	<code>yToRight=fillList(size(fillList,1)-iterFill+1);</code>
57	<code>end</code>
58	

Kode Sumber 4.8 Implementasi mendapatkan *region* menggunakan *Laplacian of Gaussian*

4.4 Implementasi Proses Segmentasi

Implementasi proses segmentasi dilakukan dengan menggunakan klasifikasi *Extreme Learning Machine* (ELM). Klasifikasi dimulai dengan tahap *training* data. Tahap ini memerlukan vektor fitur dan *ground truth* sebagai data masukan. Implementasi proses ini mengikuti diagram alir yang telah dipaparkan pada Gambar 3.17.

4.4.1 Pembangunan Data Training

Implementasi dimulai dengan membaca vektor fitur dan *ground truth*. Kemudian diambil sampel sebanyak 22.184 data pada tiap ROI atas atau bawah. Baris 10 dan 11 adalah membaca citra ROI dan *ground truth*. Kemudian citra di lakukan *preprocessing* seperti yang dipaparkan pada baris 18. Pengambilan

sampel tiap fitur dipaparkan pada baris 20 sampai 49. Terakhir, dilakukan proses klasifikasi pada kode baris 50. Penerapan pembangunan *data training* ditunjukkan pada Kode Sumber 4.9.

1	<code>addpath('Gabor_Image_features/Gabor_Image_features')</code>
2	
3	<code>maks=8;</code>
4	<code>sz=128;</code>
5	<code>dataTrain=zeros(maks*sz*sz,4);</code>
6	<code>sampel=zeros(20000,4);</code>
7	<code>z=1;</code>
8	<code>for i=1:maks</code>
9	<code>idx=num2str(i);</code>
10	<code>filenameI=strcat('Jelas2\',idx,'.bmp');</code>
11	<code>filenameGT=strcat('Jelas2\',idx,'_GT.bmp');</code>
12	<code>I=imread(filenameI);</code>
13	<code>if size(I, 3) == 3</code>
14	<code> I = rgb2gray(I);</code>
15	<code>end;</code>
16	<code>GT=imread(filenameGT);</code>
17	<code> GT=GT/255;</code>
18	<code>I=filterPrep(I,i);</code>
19	
20	<code>filter1=getGaussianFilterImage(I,sqrt(2));</code>
21	<code>filter2=getGaussianFilterImage(I,2);</code>
22	<code>filter3=imread(strcat('PC\',num2str(i),'_GT.bmp'))*255;</code>
23	<code>z=2773*(i-1)+1;</code>
24	
25	<code> if mod((i-1)*sz*sz+(j-1)*sz+k,6)==0</code>
26	<code> sampel(z,1)=GT(j,k);</code>
27	<code> sampel(z,2)=I(j,k);</code>
28	<code> sampel(z,3)=filter1(j,k);</code>

29	<code>sampel(z,4)=filter2(j,k);</code>
30	<code>sampel(z,5)=filter3(j,k);</code>
31	
32	<code>z=z+1;</code>
33	<code>size=size(dataTrain,2);</code>
34	<code>[TrainTime,OutputWeight]=elm_kernel(2,sampel,data Test,1,1,'lin_kernel',[0.1,10])</code>

Kode Sumber 4.9 Implementasi pembangunan data training

4.4.2 Proses Klasifikasi *training* menggunakan *Extreme Learning Machine*

Implementasi proses klasifikasi dimulai dengan membaca data *testing* dan sampel. Langkah selanjutnya adalah mendapatkan hasil segmentasi dari model ELM dipaparkan pada baris 2 sampai 15. Kode sumber dari proses klasifikasi untuk *training* ditunjukkan pada Kode Sumber 4.10 **Error! Reference source not found.**

1	<code>REGRESSION=0;</code>
2	<code>CLASSIFIER=1;</code>
3	
4	
5	<code>T=TrainingData_File(:,1)';</code>
6	<code>P=TrainingData_File(:,2:size(TrainingData_File,2))';</code>
7	<code>clear TrainingData_File;</code>
8	<code>disp('Load Training Data done');</code>
9	
10	<code>C = Regularization_coefficient;</code>
11	<code>NumberOfTrainingData=size(P,2);</code>
12	
13	
14	<code>number_class=kelas;</code>
15	<code>NumberOfOutputNeurons=number_class;</code>

16	
17	%%%%%%%%%% Processing the targets of training
18	temp_T=zeros (NumberofOutputNeurons, NumberofTrainingData);
19	%%%%%%%%%% TrainingPhase %%%%%%%%%%%%%%%%%%%%%%%%%%
20	disp('Training Phase');
21	tic;
22	n = size(T,2);
23	Omega_train = P'*P;
24	
25	OutputWeight=((Omega_train+speye(n)/C)\(T'));
26	TrainingTime=toc;
27	disp('->Weight done');
28	

Kode Sumber 4.10 Implementasi proses klasifikasi *training* menggunakan *Extreme Learning Machine*.

4.4.3 Proses Klasifikasi *testing* menggunakan *Extreme Learning Machine*

1	%%%%%%%%%% Calculate the output of testing input
2	disp('Testing Phase');
3	tic;
4	Omega_test = sampelTrain'*dataTest;
5	disp('->Kernel done');
6	TY=(Omega_test'* OutputWeight)'; % TY: the actual output of the testing data
7	TestingTime=toc;
8	disp('->Weight done');
9	%%%%%%%%%% Calculate training & testing classification accuracy

10	
11	<code>if Elm Type == REGRESSION</code>
12	<code>%%%%%%%%%% Calculate training & testing accuracy (RMSE) for regression case</code>
13	<code>TrainingAccuracy=sqrt (mse (T - Y)) ;</code>
14	<code>TestingAccuracy=sqrt (mse (TV.T - TY)) ;</code>
15	<code>end</code>

Kode Sumber 4.11 Implementasi proses klasifikasi *testing* menggunakan *Extreme Learning Machine*.

4.5 Implementasi *Postprocessing*

Postprocessing adalah perbaikan segmen pada citra biner. Implementasi proses ini mengikuti diagram alir yang telah dipaparkan pada Gambar 3.19. *Postprocessing* adalah perbaikan segmen pada citra biner. Operasi yang pertama kali dilakukan adalah memperhalus hasil segmentasi menggunakan filter median yang dipaparkan pada baris 1. Kemudian dilanjutkan dengan pengambilan area besar menggunakan fitur matlab *regionprops* yang dipaparkan pada baris 3 sampai 16. Operasi ini dilakukan untuk membersihkan *noise* dan mengisi *hole* dimana *noise* dan *hole* tersebut memiliki area yang bernilai kecil. Kode sumber dari proses klasifikasi ditunjukkan pada Kode Sumber 4.12.

1	<code>I=medfilt2(I,[5 5]);</code>
2	<code>%image dicomplement untuk mengisi hole</code>
3	<code>Ibw=imcomplement(I);</code>
4	<code>Ibw=logical(Ibw);</code>
5	<code>IbwL=bwlabel(Ibw);</code>
6	<code>stats=regionprops(Ibw,'Area');</code>
7	<code>choosen=find([stats.Area]>300);</code>
8	<code>imOutput=ismember(IbwL,choosen);</code>
9	<code>imOutput=imcomplement(imOutput);</code>
10	

11	<code>Ibw=imOutput;</code>
12	<code>Ibw=logical(Ibw);</code>
13	<code>IbwL=bwlabel(Ibw);</code>
14	<code>stats=regionprops(Ibw, 'Area');</code>
15	<code>choosen=find([stats.Area]>300);</code>
16	<code>imOutput=ismember(IbwL, choosen);</code>

Kode Sumber 4.12 Implementasi proses *postprocessing*

[Halaman ini sengaja dikosongkan]

BAB V

UJI COBA DAN EVALUASI

Dalam bab ini dibahas mengenai hasil uji coba sistem yang telah dirancang dan dibuat. Uji coba dilakukan untuk mengetahui kinerja sistem dengan lingkungan uji coba yang telah ditentukan serta dilakukan sesuai dengan skenario uji coba.

5.1 Lingkungan Uji Coba

Lingkungan uji coba yang digunakan adalah sebuah komputer *notebook* dengan spesifikasi sebagai berikut.

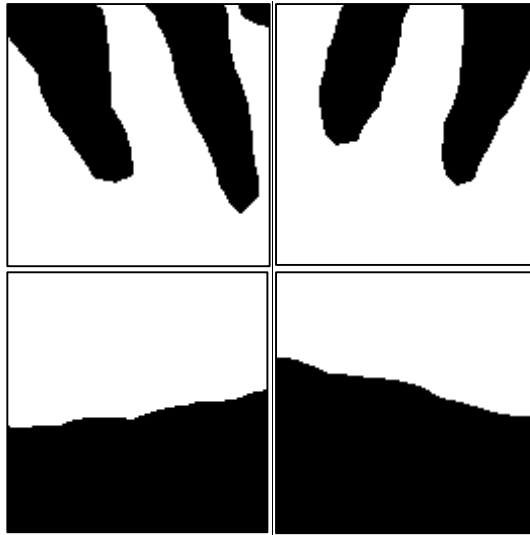
- Perangkat keras:
 - Prosesor : Intel® Core™ i3 4150 CPU @ 3,5GHz
 - Memori : 4.00 GB
- Perangkat lunak:
 - Sistem operasi : Windows 8.1
 - IDE : MATLAB R2014a

5.2 Data Uji Coba

Data uji coba yang digunakan sebagai masukan adalah citra *dental panoramic radiograph* dari 30 pasien. Dari citra *dental panoramic radiograph* dilakukan proses *cropping* untuk mendapatkan ROI pada bagian *trabecular bone* kiri dan kanan. Kemudian ROI tersebut dibagi menjadi bagian atas dan bawah. ROI dari *trabecular bone* yang dipilih ditunjukkan dengan garis putus-putus pada Gambar 5.1. Ukuran ROI yang digunakan adalah 128×128 pixel. Hasil dari proses *cropping* ditunjukkan pada Gambar 5.2.

Untuk menguji kebenaran dari hasil segmentasi, digunakan data *groundtruth* berupa citra hitam putih. Data *groundtruth* juga mengalami proses *cropping* dengan ukuran dan posisi yang sama dengan citra masukan. Data *groundtruth* ditunjukkan pada Gambar 5.3.

Gambar 5.2 Citra hasil cropping.



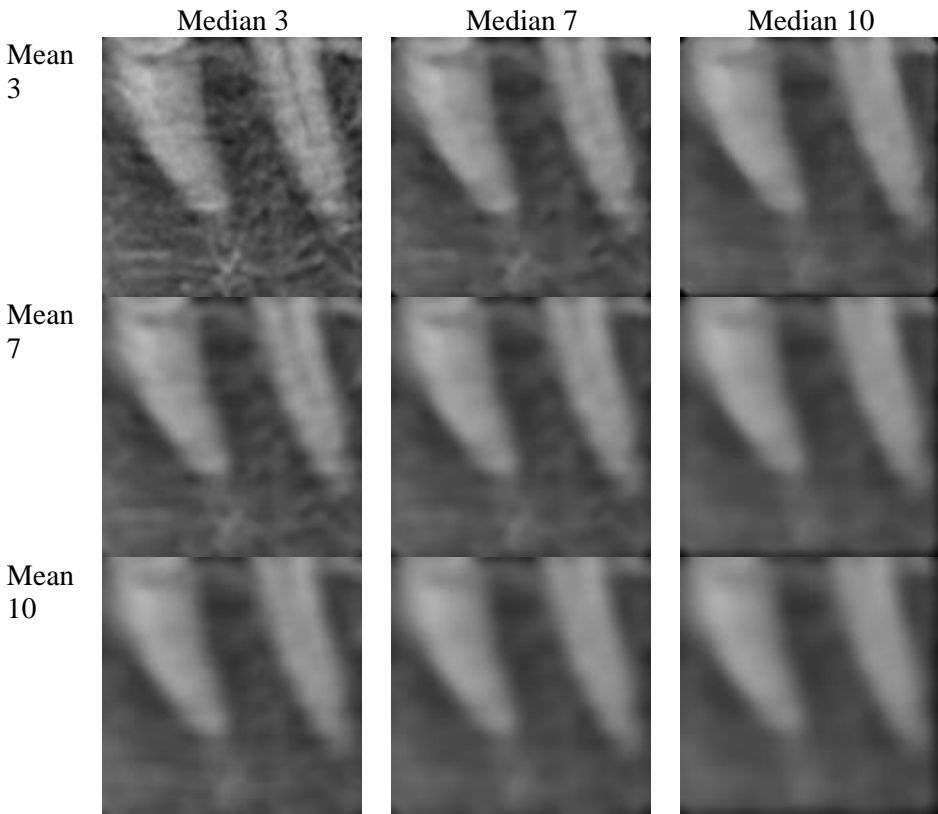
Gambar 5.3 Citra groundtruth.

5.3 Uji Coba *Preprocessing*

5.3.1 Uji Coba Penentuan Ukuran Matrik Konvolusi pada Filter Mean dan Filter Median

Uji coba ini berfokus pada pengaruh ukuran konvolusi yang berbeda pada hasil dari filter mean dan filter median. Ukuran matriks konvolusi yang digunakan adalah 3, 7, dan 10. Hasil dari uji coba filter mean dan filter median dengan ukuran matriks konvolusi yang bervariasi ditunjukkan pada Gambar 5.4.

Dari gambar tersebut disimpulkan bahwa semakin besar ukuran matrik konvolusi pada masing-masing filter, *noise* semakin hilang. Namun semakin besar ukuran matrik konvolusi pada filter mean, struktur gigi semakin besar. Sedangkan, semakin besar ukuran matriks konvolusi pada filter median, struktur gigi hampir tetap. Oleh karena itu dipilih matrik konvolusi filter mean sebesar 3 dan filter median sebesar 10.

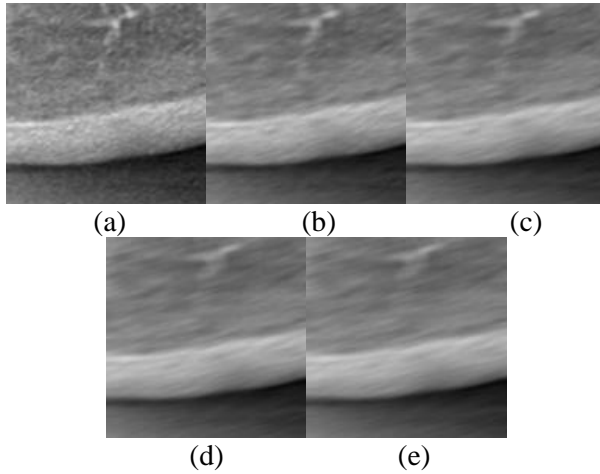


Gambar 5.4 Hasil uji coba filter Mean dan filter Median dengan ukuran matrik konvolusi masing-masing 3, 7, 10

5.3.2 Uji Coba Penentuan Ukuran Matrik Konvolusi pada filter *motion blur*

Uji coba ini berfokus pada pengaruh ukuran konvolusi yang berbeda pada hasil dari filter *motion blur*. Ukuran matriks konvolusi yang digunakan adalah 3, 7, 10, 13, dan 15. Hasil dari uji coba filter *motion blur* dengan ukuran matrik konvolusi yang bervariasi ditunjukkan pada Gambar 5.5.

Dari gambar tersebut disimpulkan bahwa semakin besar matriks konvolusi pada filter *motion blur*, *noise* semakin tidak terlihat.

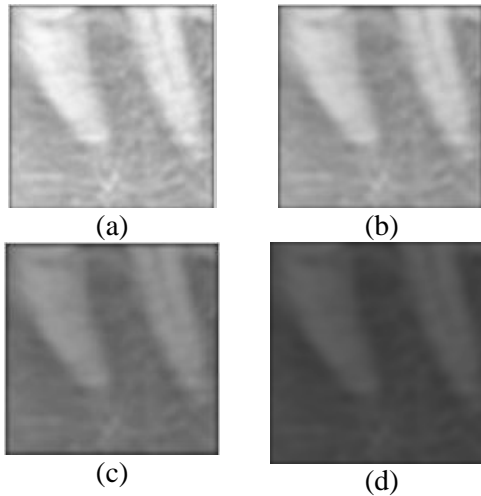


Gambar 5.5 Hasil uji coba filter *Motion Blur* dengan ukuran matrik konvolusi masing-masing (a) 3, (b) 7, (c) 10, (d) 13 dan € 15

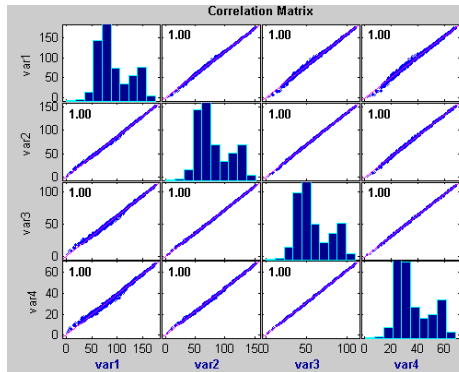
5.4 Uji Coba Ekstraksi Fitur

5.4.1 Uji Coba Penentuan Parameter Sigma pada filter *Gaussian 2D*

Uji coba ini berfokus pada pengaruh nilai sigma yang berbeda pada hasil dari filter *Gaussian 2D*. Nilai sigma yang digunakan dalam uji coba adalah $\sqrt{2}$, 2, $2\sqrt{2}$, 4. Hasil dari uji coba filter *Gaussian 2D* dengan nilai sigma yang bervariasi ditunjukkan pada Gambar 5.6. Matriks korelasi dari masing-masing citra diatas dipaparkan pada Gambar 5.7. Dari matriks korelasi tersebut disimpulkan bahwa dengan perbedaan sigma, hasil filter *Gaussian* tidak memiliki perbedaan yang cukup signifikan.



Gambar 5.6 Hasil uji coba filter Gaussian 2D menggunakan sigma (a) $\sqrt{2}$ (b) 2 (c) $2\sqrt{2}$ (d) 4

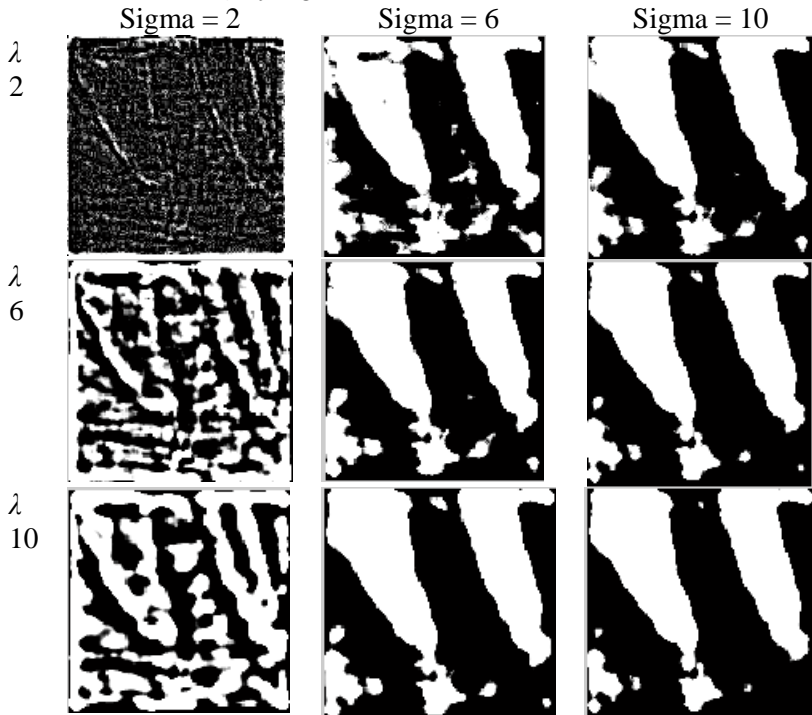


Gambar 5.7 Matrik korelasi antara hasil filter Gaussian 2D dengan sigma $\sqrt{2}$, 2, $2\sqrt{2}$, 4.

5.4.2 Uji Coba Penentuan Parameter Sigma dan Wavelength pada Metode Log Gabor

Uji coba ini berfokus pada pengaruh nilai sigma dan wavelength yang berbeda pada hasil dari filter *Log Gabor*. Nilai sigma dan wavelength (λ) yang digunakan dalam uji coba adalah 2,

6, 10. Hasil dari uji coba metode filter *Log Gabor* dengan nilai sigma dan *wavelength* (λ) yang berbeda ditunjukkan pada Gambar 5.8. Dari gambar tersebut terlihat bahwa semakin besar nilai sigma, maka *fill* dari citra semakin kuat dan *noise* dari citra tersebut semakin hilang. Sedangkan semakin besar nilai *wavelength* semakin lebar *edge* yang diambil.



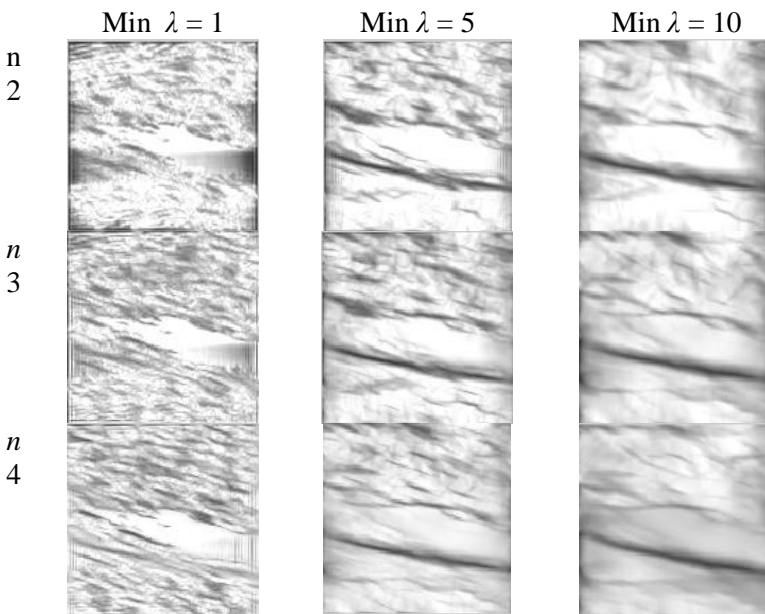
Gambar 5.8 Hasil uji coba filter Log Gabor
menggunakan Sigma dan Wavelength yang berbeda

5.4.3 Uji Coba Penentuan Nilai Skala Wavelet dan *Wavelength* Awal pada *Phase Congruency*

Uji coba ini berfokus pada pengaruh nilai skala *Wavelet* (n) dan *wavelength* (λ) awal yang berbeda pada hasil dari metode

Phase Congruency. Nilai skala *Wavelet* yang diujikan adalah 2,3, dan 4. Nilai *wavelength* awal yang diujikan adalah 1,5, dan 10. Hasil dari uji coba metode *Phase Congruency* dengan nilai skala *Wavelet* (n) dan *wavelength* (λ) awal yang berbeda ditunjukkan pada Gambar 5.9 dengan meng-*complement* citra agar tepi terlihat lebih jelas.

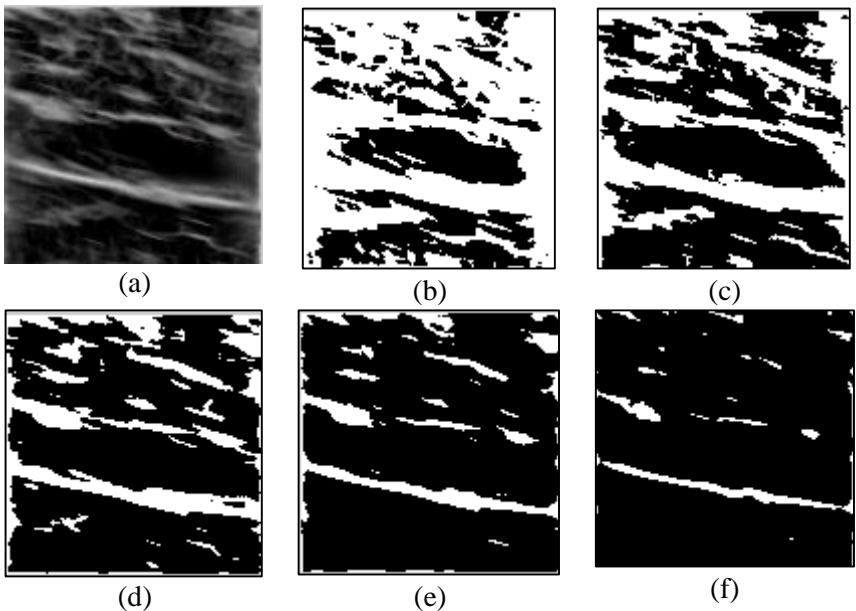
Dari gambar tersebut terlihat bahwa nilai panjang *wavelength* awal menentukan jumlah serat proses. Pada *wavelength* awal 1 tidak dapat terlihat batas *cortical bone*. Pada *wavelength* awal 10 terdapat batas atas *cortical bone* yang putus. *wavelength* awal terbaik pada uji coba ini didapatkan sebesar 5. Sedangkan dari segi banyak skala wavelet tidak terlalu terlihat pengaruhnya.



Gambar 5.9 Hasil uji coba *phase congruency* menggunakan nilai Sigma dan *Wavelength* yang berbeda

5.4.4 Uji Coba Penentuan Nilai *Threshold* Binerisasi Hasil *Phase Congruency*

Uji coba ini berfokus pada pengaruh nilai *threshold* binerisasi hasil *phase congruency* untuk mendapatkan batas pada *cortical bone*. Nilai batas binerisasi yang diujikan adalah 0,1; 0,2; 0,3; 0,4; dan 0,5. Hasil dari uji coba penentuan batas binerisasi yang berbeda ditunjukkan pada Gambar 5.10. Pada batas 0,1 sampai 0,2, serat yang terbentuk terlalu banyak sehingga tidak didapatkan *edge cortical bone*. Sedangkan pada batas 0,4 dan 0,5, serta pada *cortical bone* atas tidak tersambung secara merata, sehingga didapatkan batas optimal yaitu 0,3.



Gambar 5.10 Hasil uji coba binerisasi *phase congruency*
(a) citra asli, binerisasi (b) 0.1 (c) 0.2 (d) 0.3 (e) 0.4 (f) 0.5

5.5 Uji Coba Kinerja Sistem Segmentasi

Pada tahap ini tiap piksel hasil klasifikasi dibandingkan dengan tiap piksel pada data *groundtruth*. Proses ini dilakukan dengan menggunakan *confusion matrix*. Dari *confusion matrix* didapatkan nilai *true positive*, *true negative*, *false positive* dan *false negative*. Penjelasan mengenai nilai-nilai tersebut ditunjukkan pada tabel Tabel 5.1.

Dari nilai-nilai tersebut dapat dihitung tingkat akurasi, *sensitivity* dan *specificity*. Tingkat akurasi berfungsi sebagai parameter kebenaran keseluruhan pixel hasil klasifikasi. Perhitungan tingkat akurasi dilakukan dengan membagi nilai hasil penjumlahan TP dan TN dengan jumlah keseluruhan pixel. Perhitungan akurasi ditunjukkan pada Persamaan 5.1.

Tabel 5.1 Hasil Confusion Matrix

	<i>Trabecular</i> pada citra hasil segmentasi	<i>Non-trabecular</i> pada citra hasil segmentasi
<i>Trabecular</i> pada citra <i>Ground-Truth</i>	<i>True Positive</i> (TP)	<i>True Negative</i> (TN)
<i>Non-trabecular</i> pada citra <i>Ground Truth</i>	<i>False Positive</i> (FP)	<i>False Negative</i> (FN)

True positive (TP) adalah pixel *trabecular* yang diklasifikasi secara benar sebagai pixel *trabecular*. *True negative* (TN) adalah pixel bukan *trabecular* yang diklasifikasi secara benar sebagai pixel bukan *trabecular*. *True positive* (FP) adalah pixel *trabecular* yang diklasifikasi secara salah sebagai pixel bukan *trabecular*. *True negative* (FN) adalah pixel bukan *trabecular* yang diklasifikasi secara salah sebagai pixel *trabecular*.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \quad (5.1)$$

Karena perbandingan jumlah piksel *trabecular* dan bukan *trabecular* tidak seimbang, maka diperlukan perhitungan tingkat *sensitivity* dan *specificity*. *Sensitivity* berfungsi sebagai parameter kebenaran hasil klasifikasi untuk piksel *trabecular*. *Specificity* berfungsi sebagai parameter kebenaran hasil klasifikasi untuk piksel bukan *trabecular*. *Sensitivity* didapatkan dengan membagi nilai TP dengan penjumlahan nilai TP dan FP. Perhitungan *sensitivity* ditunjukkan pada Persamaan 5.2. Sedangkan *specificity* didapatkan dengan membagi nilai TN dengan penjumlahan nilai TN dan FN. Perhitungan *specificity* ditunjukkan pada Persamaan 5.3

$$Sensitivitas = \frac{TP}{TP+FP} \quad 5.2$$

$$Spesifisitas = \frac{TN}{TN+FN} \quad 5.3$$

Uji kinerja diterapkan pada ROI atas dan ROI bawah dengan metode klasifikasi *Extreme Learning Machine* yang digunakan dalam proses segmentasi. Pengujian kinerja ini menghitung akurasi, sensitivitas, spesifisitas, dan waktu eksekusi yang dibutuhkan untuk segmentasi tiap citra.

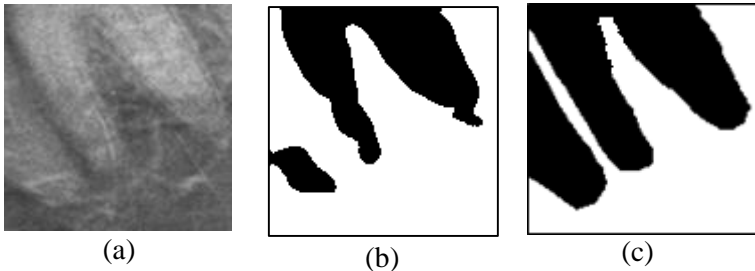
5.5.1 ROI atas

Hasil pengujian pada ROI atas ditunjukkan pada Tabel 5.2. Pengujian menggunakan data masukan sebanyak 20 citra. ROI *trabecular* atas pada masing-masing citra didapatkan rata-rata sensitivitas, spesifisitas, akurasi, dan waktu eksekusi masing-masing 82,31 %, 93,67 %, 90,33 %, dan 2,59 detik.

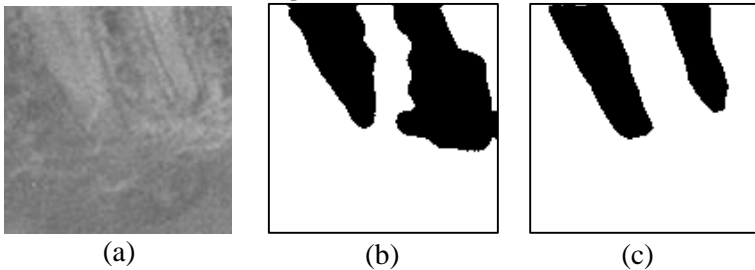
Pada kolom sensitivitas, nilai tertinggi didapatkan dari citra 1 bagian kanan dengan nilai 99,3%, sedangkan nilai terendah didapatkan dari citra 6 kanan dengan nilai 56,8%. Citra 1 bagian kanan dan citra 6 bagian kanan ditunjukkan pada Gambar 5.11 dan Gambar 5.12.

Tabel 5.2 Hasil pengujian segmentasi ROI atas

Citra	Sensitivitas (%)		Spesifisitas (%)		Akurasi (%)		Waktu Segmentasi (detik)	
	Kanan	Kiri	Kanan	Kiri	Kanan	Kiri	Kanan	Kiri
1	99,3	87,6	73,8	89,7	82,2	89,0	2,56	2,68
2	86,9	76,2	94,7	97,7	92,3	92,7	2,64	2,73
3	59,9	95,7	98,0	94,3	86,6	94,7	2,58	2,56
4	92,9	10,0	94,1	72,2	93,7	80,8	2,65	2,56
5	97,9	96,0	91,1	96,3	93,2	96,2	2,56	2,61
6	56,8	94,3	96,8	92,0	84,1	92,7	2,52	2,57
7	70,2	62,4	97,3	97,7	87,0	86,1	2,55	2,56
8	83,3	85,6	98,3	98,1	94,1	95,2	2,62	2,52
9	80,2	96,9	96,4	96,1	91,2	96,3	2,55	2,63
10	93,1	89,7	98,2	95,1	97,2	93,2	2,56	2,53
11	94,8	64,5	96,8	97,6	96,3	89,2	2,62	2,59
12	73,8	67,2	97,5	99,8	92,5	90,1	2,58	2,59
13	97,8	90,6	84,9	93,0	89,3	92,1	2,57	2,58
14	81,2	88,1	97,5	95,9	92,7	93,6	2,58	2,66
15	90,2	89,4	85,0	95,6	86,6	93,7	2,61	2,60
16	70,6	61,2	83,4	93,6	79,2	81,5	2,61	2,55
17	86,6	73,5	92,9	96,0	90,9	89,6	2,57	2,65
18	61,1	79,0	95,2	89,0	85,5	85,9	2,56	2,56
19	94,8	64,5	96,8	97,6	96,3	89,2	2,64	2,54
20	90,1	68,5	92,5	98,5	91,8	89,0	2,60	2,64
Rata-rata	82,31		93,67		90,33		2,59	



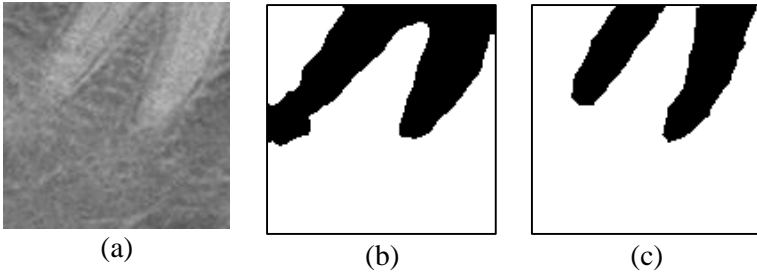
**Gambar 5.11 Citra dengan performa sensitivitas tertinggi
Citra 1 kanan (a) citra asli (b) citra hasil segmentasi (c) citra
groundtruth.**



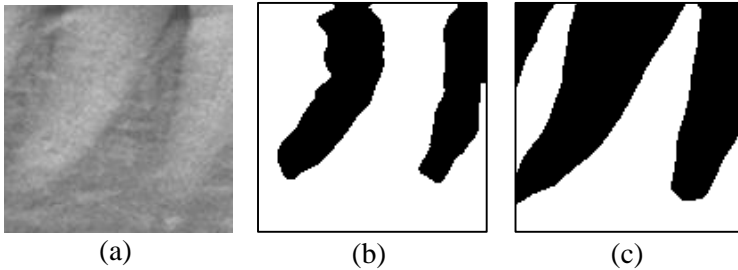
**Gambar 5.12 Citra dengan performa sensitivitas terendah
Citra 6 kanan (a) citra asli (b) citra hasil segmentasi (c) citra
groundtruth**

Pada kolom spesifisitas, nilai tertinggi didapatkan dari citra 12 bagian kiri dengan nilai 99,8%, sedangkan nilai terendah didapatkan dari citra 4 bagian kiri dengan nilai 72.2%. Citra 12 bagian kiri dan citra 4 bagian kiri ditunjukkan pada Gambar 5.13 dan Gambar 5.14.

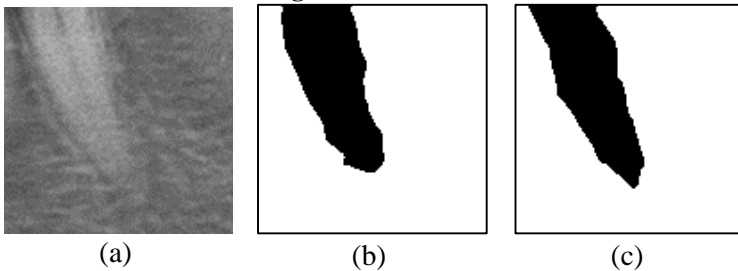
Pada kolom akurasi, nilai tertinggi didapatkan dari citra 10 bagian kanan dengan nilai 97.2%, sedangkan nilai terendah didapatkan dari citra 4 bagian kiri dengan nilai 80,8%. Citra 10 bagian kanan dan citra 4 bagian kiri ditunjukkan pada



Gambar 5.13 Citra dengan performa spesifisitas tertinggi
Citra 12 bagian kiri (a) citra asli (b) citra hasil segmentasi
(c) citra groundtruth.



Gambar 5.14 Citra dengan performa spesifisitas dan akurasi terendah
Citra 4 bagian kiri (a) citra asli (b) citra hasil segmentasi (c) citra
groundtruth.



Gambar 5.15 Citra dengan performa akurasi tertinggi
Citra 10 bagian kanan (a) citra asli (b) citra hasil segmentasi
(c) citra groundtruth.

Tabel 5.3 Hasil pengujian segmentasi ROI bawah

Citra	Sensitivitas (%)		Spesifisitas (%)		Akurasi (%)		Waktu Segmentasi (detik)	
	Kanan	Kiri	Kanan	Kiri	Kanan	Kiri	Kanan	Kiri
1	99,9	99,5	94,3	94,5	96,9	97,4	3,05	2,62
2	99,6	98,4	95,7	97,3	98,0	97,7	2,53	2,58
3	99,9	99,8	98,0	97,6	99,0	98,8	2,57	2,61
4	95,8	62,4	98,9	100,0	97,2	67,8	2,62	2,60
5	99,6	97,1	98,7	98,0	99,1	97,5	2,57	2,54
6	90,5	95,1	99,4	98,5	95,5	96,9	2,56	2,57
7	99,9	99,9	97,9	97,8	98,7	98,8	2,54	2,50
8	80,5	99,9	98,5	98,6	87,9	99,4	2,53	2,55
9	99,1	99,0	94,9	98,5	96,6	98,6	2,54	2,57
10	82,4	79,6	100,0	100,0	90,8	89,6	2,71	2,57
11	98,9	99,8	94,9	97,0	97,2	98,6	2,66	2,57
12	99,7	99,7	98,0	95,3	98,9	98,0	2,53	2,71
13	99,8	76,8	97,7	50,8	98,4	67,0	2,67	2,62
14	99,7	99,9	97,3	98,0	98,7	98,9	2,57	2,56
15	98,0	99,8	98,3	97,2	98,2	99,0	2,55	2,54
16	99,9	98,9	98,2	97,3	99,2	98,1	2,63	2,55
17	98,4	85,8	96,6	99,7	97,5	91,5	2,54	2,53
18	99,8	99,7	98,3	97,9	99,2	98,7	2,54	2,66
19	98,9	99,8	94,9	97,0	97,2	98,6	2,66	2,62
20	70,7	98,6	100,0	98,4	78,3	98,5	2,57	2,53
Rata-rata	95,01		96,50		95,29		2,59	

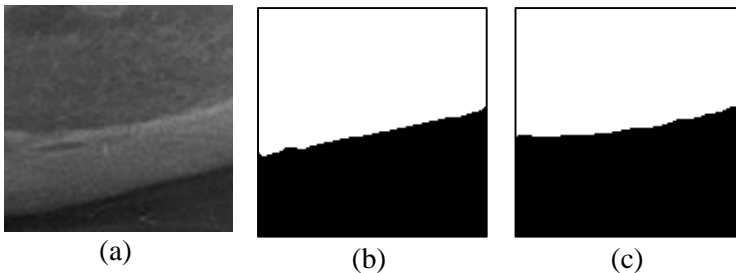
5.5.2 ROI bawah

Hasil pengujian pada ROI bawah ditunjukkan pada Tabel 5.3. Pengujian menggunakan data masukan sebanyak 20 citra. ROI *trabecular* bawah pada masing-masing citra didapatkan rata-rata sensitivitas, spesifisitas, akurasi, dan waktu eksekusi masing-masing 95,01%, 96,50%, 95,29% dan 2,59 detik.

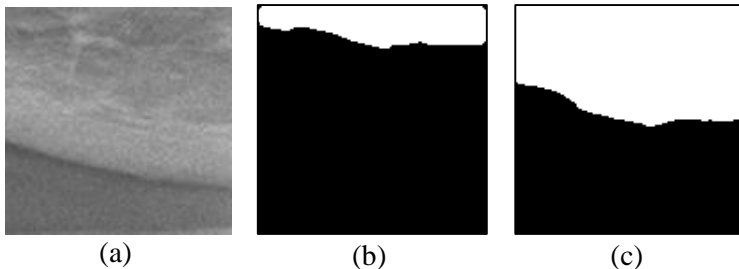
Pada kolom sensitivitas, nilai tertinggi didapatkan dari citra 1 bagian kanan, citra 3 bagian kanan, dan citra 7 bagian kanan, dengan nilai 99,9%, sedangkan nilai terendah didapatkan dari citra 4 bagian kiri dengan nilai 44,13%. Citra 1 bagian kanan dan citra 4 bagian kiri ditunjukkan pada Gambar 5.16 dan Gambar 5.17.

Pada kolom spesifisitas, nilai tertinggi didapatkan dari citra 4 bagian kiri, citra 10 bagian kanan dan kiri dengan nilai 100%, sedangkan nilai terendah didapatkan dari citra 13 bagian kiri dengan nilai 50,8%. Citra 4 bagian kiri dan citra 13 bagian kiri ditunjukkan pada Gambar 5.18 dan Gambar 5.19.

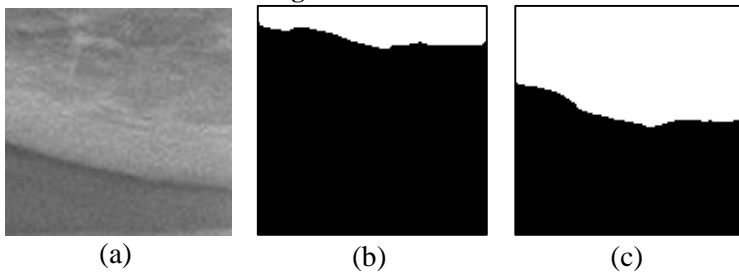
Pada kolom akurasi, nilai tertinggi didapatkan dari citra 8 bagian kiri dengan nilai 99,4%, sedangkan nilai terendah didapatkan dari citra 13 bagian kiri dengan nilai 67,0%. Citra 8 bagian kiri dan citra 13 bagian kiri ditunjukkan pada Gambar 5.20 dan Gambar 5.19.



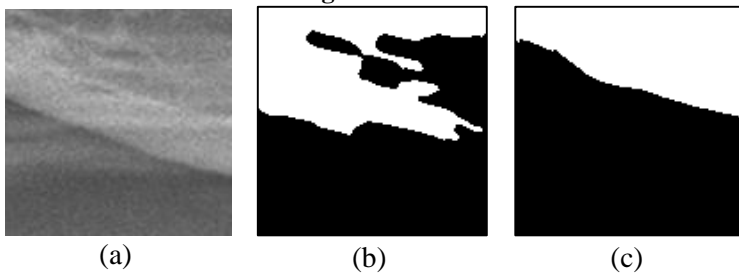
Gambar 5.16 Citra dengan performa sensitivitas tertinggi
Citra 1 bagian kanan (a) citra asli (b) citra hasil segmentasi
(c) citra groundtruth.



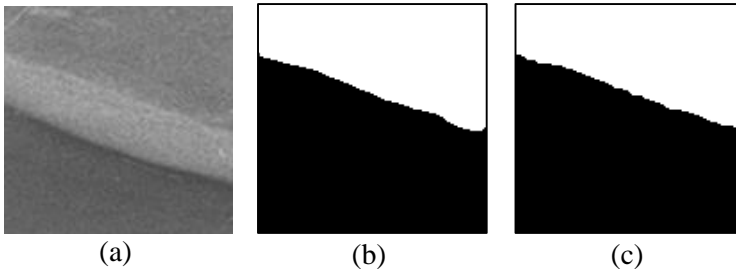
**Gambar 5.17 Citra dengan performa sensitivitas terendah
Citra 4 bagian kiri (a) citra asli (b) citra hasil segmentasi (c) citra
groundtruth.**



**Gambar 5.18 Citra dengan performa spesifisitas tertinggi
Citra 4 bagian kiri (a) citra asli (b) citra hasil segmentasi (c)
citra groundtruth.**



**Gambar 5.19 Citra dengan performa spesifisitas terendah
dan akurasi terendah
Citra 13 bagian kiri (a) citra asli (b) citra hasil segmentasi (c)
citra groundtruth.**



Gambar 5.20 Citra dengan performa akurasi tertinggi
Citra 8 bagian kiri (a) citra asli (b) citra hasil segmentasi (c)
citra groundtruth.

5.6 Evaluasi

Pada subbab ini akan dijelaskan hasil dari serangkaian uji coba yang dilakukan. Selain itu dijelaskan juga kendala yang dihadapi selama proses pengerjaan.

Tiga metode yang diuji coba pada *preprocessing* antara lain: filter mean, filter median, dan filter *motion blur*. Pada proses ekstraksi fitur ada tiga metode yang diuji coba, yaitu filter *Gaussian* 2D, Filter *Log Gabor* dan *phase congruency*. Pengujian dilakukan dengan menguji salah satu parameter yang digunakan. Tidak semua parameter diuji karena keterbatasan waktu. Diharapkan pada penelitian selanjutnya dapat dilakukan pengujian pada parameter-parameter tersebut.

Uji coba filter mean dan filter median berfokus pada pengaruh ukuran konvolusi yang berbeda pada hasil dari filter mean dan filter median. Ukuran matrik konvolusi yang digunakan adalah 3, 7, dan 10. Dari pengujian diketahui bahwa semakin besar ukuran matrik konvolusi pada masing-masing filter, *noise* semakin hilang. Namun semakin besar ukuran matrik konvolusi pada filter mean, struktur gigi semakin besar. Sedangkan, semakin besar ukuran matriks konvolusi pada filter median, struktur gigi hampir tetap.

Uji coba filter *motion blur* berfokus pada pengaruh ukuran konvolusi yang berbeda pada hasil dari filter *motion blur*. Ukuran

matrik konvolusi yang digunakan adalah 3, 7, 10, 13, dan 15. Dari pengujian diketahui bahwa semakin besar matriks konvolusi pada filter *motion blur*, *noise* semakin tidak terlihat.

Pengujian filter *Gaussian* 2D dilakukan untuk mengetahui pengaruh nilai sigma pada hasil ekstraksi fitur. Dari pengujian diketahui bahwa nilai sigma tidak berpengaruh signifikan pada hasil ekstraksi fitur.

Pengujian metode *Log Gabor* dilakukan untuk mengetahui pengaruh nilai sigma dan *wavelength* pada hasil ekstraksi fitur. Dari pengujian didapatkan bahwa semakin besar nilai sigma, maka *fill* dari citra semakin kuat dan *noise* dari citra tersebut semakin hilang. Sedangkan semakin besar nilai *wavelength* semakin lebar *edge* yang diambil. Maka dari itu, dipilih nilai 10 pada sigma dan *wavelength*.

Pengujian metode *Phase Congruency* dilakukan untuk mengetahui pengaruh nilai skala *Wavelet* dan *Wavelength* awal pada hasil ekstraksi fitur. Dari pengujian diketahui bahwa nilai panjang *wavelength* awal menentukan jumlah serat proses. Pada *wavelength* awal 1 tidak dapat terlihat batas *cortical bone*. Pada *wavelength* awal 10 terdapat batas atas *cortical bone* yang putus. *wavelength* awal terbaik pada uji coba ini didapatkan sebesar 5. Sedangkan dari segi banyak skala *wavelet* tidak terlalu terlihat pengaruhnya.

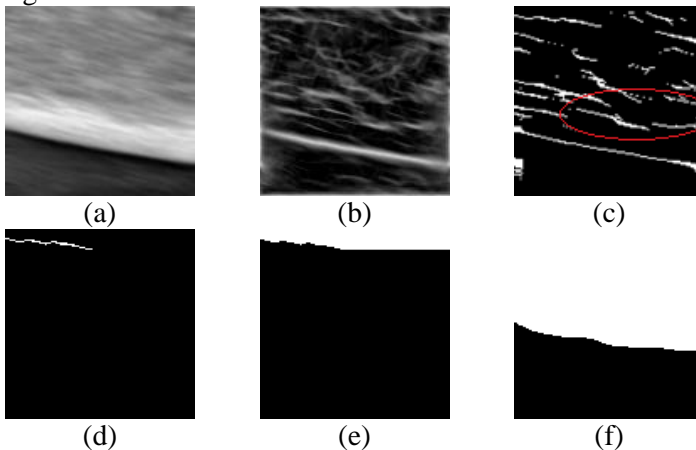
Pengujian *Laplacian of Gaussian* tidak diperlukan karena menggunakan format *default* dari matlab yaitu agar didapa.

Pengujian binerisasi untuk mendapatkan segmen *trabecular bone* menggunakan *phase congruency* dilakukan untuk mengetahui pengaruh nilai batas binerisasi pada hasil ekstraksi fitur. Dari pengujian didapatkan batas sebesar 0,3 memberikan hasil yang terbaik dibanding batas uji lainnya sehingga nilai tersebut dipilih sebagai parameter tetap pada proses binerisasi hasil *phase congruency*.

Pada ekstraksi fitur pengambilan intensitas piksel tidak diperlukan uji coba karena tidak ada parameter yang digunakan. Selain pengujian metode ekstraksi fitur, pengujian juga dilakukan

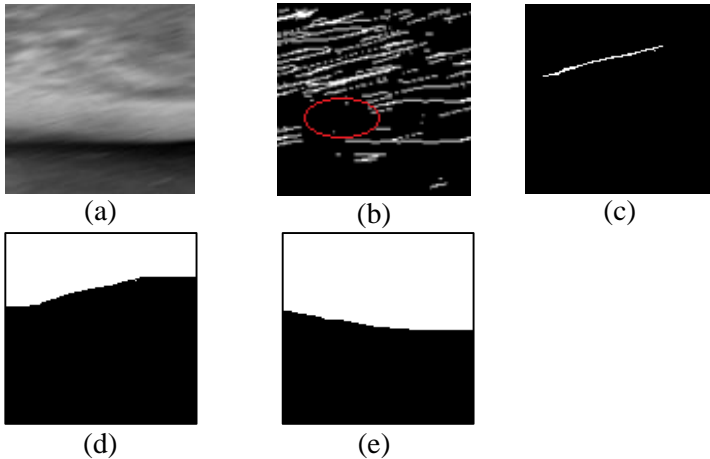
untuk metode segmentasi. Ada dua ROI yang diuji coba, yaitu ROI atas dan ROI bawah. Dari hasil uji coba, ROI atas memiliki nilai rata-rata akurasi, *specificity* dan *sensitivity* di atas 80%. Sedangkan ROI bawah memiliki nilai rata-rata akurasi, *specificity* dan *sensitivity* di atas 90%.

Terdapat beberapa ROI yang belum berhasil mendapatkan segmentasi yang baik karena batas antara *cortical* dan *trabecular bone* kurang terlihat jelas. Hal ini menyebabkan garis yang terdeteksi pada proses pemilihan garis baik pada *phase congruency* dan *Laplacian of Gaussian* adalah *noise* atau batas antara *cortical bone* dengan latar belakang. Hasil segmentasi yang mengambil *noise* dipaparkan pada Gambar 5.21 dan Gambar 5.22 dimana masing-masing citra tersebut adalah citra 10 bagian kiri dan citra 20 bagian kanan.



Gambar 5.21 Citra cortical bone yang terputus pada hasil *phase congruency*

Citra 10 bagian kiri (a) Citra Asli (b) Citra Phase Congruency (c) hasil binerisasi menunjukkan batas atas yang terputus (d) batas atas yang terambil adalah noise (e) citra segmentasi (f) citra groundtruth



Gambar 5.22 Citra cortical bone yang terputus pada *LoG* Citra 20 bagian kanan (a) Citra Asli (b) Citra *LoG* menunjukkan batas atas yang terputus (c) batas atas yang terambil adalah noise (d) citra segmentasi (e) citra groundtruth.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini dibahas kesimpulan dari perancangan, implementasi, uji coba, dan evaluasi sistem. Selain itu, dibahas pula saran untuk mendapatkan hasil yang lebih baik pada penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan yang diperoleh dari uji coba dan evaluasi adalah sebagai berikut:

1. Tujuan tugas akhir ini berhasil mendapatkan hasil segmentasi struktur *trabecular bone* pada *dental panoramic radiograph* berbasis karakteristik profil segmen menggunakan metode *Extreme Learning Machine*. Pada ROI atas, didapatkan rata-rata sensitivitas, spesifisitas, akurasi, dan waktu eksekusi masing-masing sebesar 82,31 %, 93,67 %, 90,33 %, dan 2,59 detik. Sedangkan pada ROI bawah didapatkan rata-rata sensitivitas, spesifisitas, akurasi, dan waktu eksekusi masing-masing sebesar 95,01%, 96,50%, 95,29% dan 2,59 detik.
2. *Preprocessing* yang digunakan pada tugas akhir ini adalah filter mean dan median. Parameter yang digunakan adalah filter mean menggunakan matriks konvolusi 3x3 dan filter median menggunakan matriks konvolusi 10x10. Namun pada ROI bawah, digunakan filter *motion blur*. Parameter yang digunakan adalah matriks konvolusi berukuran 15x15 dan arah orientasi 15 derajat untuk ROI kanan dan -15 derajat untuk ROI kiri.
3. Ekstraksi fitur yang digunakan pada ROI atas yaitu intensitas citra, filter *Gaussian 2D* menggunakan sigma $\sqrt{2}$ dan 2, filter *Log Gabor* menggunakan sigma sebesar 10 dan *wavelength* sebesar 10, *phase congruency* menggunakan nilai skala wavelet sebesar 2, *wavelength* minimum sebesar 5, dan *threshold* binerisasi sebesar 0,3, dan *Laplacian of*

Gaussian. Ekstraksi fitur ini cukup baik untuk mensegmentasi ROI *trabecular* atas.

4. Metode klasifikasi *Extreme Learning Machine* berhasil diterapkan untuk membuat segmentasi *trabecular bone*.

6.2 Saran

Saran yang dapat diberikan dalam pengujian metode perbaikan warna adalah sebagai berikut:

1. Perlu adanya perbaikan pada penentuan *threshold* binerisasi pada penentuan *region* hasil *phase congruency* untuk mendapatkan garis tepi yang baik terutama batas *cortical bone* dan *trabecular bone*.
2. Perlu adanya metode yang lebih baik lagi untuk mendeteksi batas antara *cortical bone* dengan *trabecular bone* dikarenakan pada beberapa citra memiliki kontras yang berbeda sehingga *cortical bone* hampir mirip dengan *trabecular bone*.

DAFTAR PUSTAKA

- [1] A. Z. Arifin, A. Asano, A. Taguchi, T. Nakamoto, M. Ohtsuka, M. Tsuda, Y. Kudo, K. Tanimoto, "Computer-aided System for Measuring the Mandibular Cortical Width on Dental Panoramic Radiographs in Identifying Postmenopausal Women with Low Bone Mineral Density," *Osteoporosis International*, vol. 17, pp. 753-759, 2006.
- [2] L. R. Dewi, A. Z. Arifin, A. Yuniarty, "Segmentasi Trabecular Bone Berdasarkan Linear Structure pada Citra Dental Panoramic Radiographs," pp. 1-5, 2009.
- [3] N. Yuniarti, A. Z. Arifin, M. Aryuni, "Informasi Tekstural untuk Identifikasi Trabecular Bone pada Citra Dental Panoramic Radiograph," pp. 1-5, 2008.
- [4] C. Zhu, B. Zou, R. Zhao, J. Cui, X. Duanb, Z. Chen, Y. Liang, "Retinal vessel segmentation in colour fundus images using Extreme Learning Machine," *Computerized Medical Imaging and Graphics*, 2016.
- [5] NDE/NDT Resource Center, "History," [Online]. Available: <https://www.nde-ed.org/EducationResources/CommunityCollege/Radiography/Introduction/history.htm>. [Accessed 6 February 2015].
- [6] ADA, "X-Rays," [Online]. Available: <http://www.ada.org/en/member-center/oral-health-topics/x-rays#FAQ>. [Accessed 6 February 2015].
- [7] Drotterholt, [Online]. Available: http://www.drotterholt.com/docs/Kodak_Panoramic_Radiography_Tips.pdf. [Accessed March 2015].
- [8] Kalbe Nutritionals, "Jaringan Tulang," [Online]. Available: http://www.kalbenutritionals.com/article_detail.asp?strlang=ind&id=149. [Accessed 21 February 2015].
- [9] S. N. Syarifuddin, "Analisis Filtering Citra dengan Metode," pp. 1-6, 2006.

- [10] L. Vandevenne, "Image Filtering," Lode's Computer Graphics Tutorial, 2004. [Online]. Available: <http://lodev.org/cgtutor/filtering.html>. [Accessed 20 12 2016].
- [11] T. Lindeberg, "Edge detection and ridge detection with automatic scale," *Int. J. Comput. Vis*, no. 30, p. 117–156, 1998..
- [12] D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *J. Opt. Soc. Am. A*, vol. 4, no. 12, pp. 2379-2394, 1987.
- [13] P. Kovesei, "Image features from phase congruency," *Computer Vision*, pp. 1-26, 1995.
- [14] R. Wang, "Laplacian of Gaussian (LoG)," 16 10 2016. [Online]. Available: <http://fourier.eng.hmc.edu/e161/lectures/gradient/node8.html>. [Accessed 1 19 2017].
- [15] Huang, G., Huang, G.B., Song, S., You, K. , "Trends in extreme learning machines: A review," *Neural Networks*, vol. 61, pp. 32-48, 2015.

BIODATA PENULIS



Rizqi Okta Ekoputris dilahirkan pada tanggal 13 Oktober di Jember pada tahun 1995. Pendidikan dasar hingga menengah diselesaikan di Jember, kemudian pada tahun 2013 hingga saat ini melanjutkan pendidikan tinggi di Jurusan Teknik Informatika ITS.

Penulis memiliki beberapa pengalaman organisasi dan kepanitiaan semasa kuliah diantaranya Staf Departemen Riset dan Teknologi HMTTC ITS (2014-2015), Staf Departemen Keilmuan Jamaah Masjid Manarul Ilmi ITS (JMMI ITS) (2014-2015), Staf Kepala Divisi Keilmiah Departemen Riset dan Teknologi HMTTC ITS (2015-2016), Staf Departemen Keilmuan Keluarga Muslim Informatika ITS (2015-2016), Anggota Tim Soal NLC Schematics ITS 2014, Ketua Tim Soal NLC Schematics ITS 2015. Penulis juga aktif sebagai administrator Laboratorium Komputasi Cerdas dan Visi Teknik Informatika ITS periode 2015 hingga 2016. Selain itu, penulis pernah meraih Medali Emas Data Mining Gemastik 9 2016 di Universitas Indonesia. Penulis dapat dihubungi melalui rizqi.okta@gmail.com

[Halaman ini sengaja dikosongkan]